NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

THREE-DIMENSIONAL FEATURE RECONSTRUCTION WITH DUAL FORWARD LOOKING SONARS FOR UNMANNED UNDERWATER VEHICLE NAVIGATION

by

Nevin A. McChesney

March 2009

Thesis Advisor:	Doug Horner
Thesis Co-Advisor:	Roberto Cristi

Approved for public release: distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Three-Dimensional Feature Reconstruction with Dual Forward Looking Sonars for Unmanned Underwater Vehicle Navigation		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Nevin A. McChesney			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release: distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

Unmanned Underwater Vehicles frequently rely on two-dimensional sensors for information about their surroundings. These sensors do not provide adequate information for obstacle avoidance in cluttered maritime environments. To address that issue, a three-dimensional reconstruction of the environment utilizing occupancy grids and a prototype forward looking sonar will be considered. Providing the vehicle with three-dimensional views of the environment will allow for optimal route planning and an increase in successful missions in complex environments.

14. SUBJECT TERMS Unmanned Vehicles, Forward Looking Sonar, REMUS, UUV, Occupancy Grid, 3D Reconstruction, Three-Dimensional Mapping			**15. NUMBER OF PAGES** 83
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release: distribution is unlimited

THREE-DIMENSIONAL FEATURE RECONSTRUCTION WITH DUAL FORWARD LOOKING SONARS FOR UNMANNED UNDERWATER VEHICLE NAVIGATION

Nevin A. McChesney
Lieutenant, United States Navy
B.S., University of Kansas, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2009

Author: Nevin A. McChesney

Approved by: Doug Horner
Thesis Advisor

Roberto Cristi
Thesis Co-Advisor

Jeffrey Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Unmanned Underwater Vehicles frequently rely on two-dimensional sensors for information about their surroundings. These sensors do not provide adequate information for obstacle avoidance in cluttered maritime environments. To address that issue, a three-dimensional reconstruction of the environment utilizing occupancy grids and a prototype forward looking sonar will be considered. Providing the vehicle with three-dimensional views of the environment will allow for optimal route planning and an increase in successful missions in complex environments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Unmanned underwater vehicles (UUV) are being tasked with a growing number of mission sets. These mission sets require the vehicle to perform tasks in a wide variety of environments. The mission sets may require the UUV to traverse up river ways, perform searches in harbor environments, or navigate in littoral areas. In each of these areas, the complexity of the environment poses challenges for the safe navigation of the vehicle.

The current build of UUVs are utilizing sensors that provide two-dimensional representations of their operating environment. In compressing the information from a three-dimensional world into a two-dimensional representations a significant amount of information is lost. The loss of this information creates challenges in the proper navigation of the vehicles. The main goal of this work is to provide the vehicle with a three-dimensional representation of the environment that can be utilized for obstacle avoidance and route planning.

With this goal in mind, forward looking sonar (FLS) was chosen as the sensor of interest. This research mounted two prototype FLS onto the Remote Environmental Measuring UnitS (REMUS) used by the Center for Autonomous Vehicle Research (CAVR) at the Naval Postgraduate School. One of the sensors provided a horizontal field of view that covered ninety degrees and ninety meters in front of the vehicle. The horizontal FLS also has fifteen degrees of ambiguity in the vertical direction. The other FLS was mounted to provide a vertical field of view that covers forty-five degrees and ninety meters, with fifteen degrees of ambiguity in the horizontal direction.

An occupancy grid was used as the framework of the reconstruction. Occupancy grids divide the area into sub regions. Each region or cell is assigned a probability that indicates the likelihood of that cell being occupied. The probabilities of occupancy are determined from physical and probabilistic models of the sensors. Each measurement received by the sensors is converted into a probability of occupancy and provides an

update to the overall probability stored in the grid. With enough measurements, a clear three-dimensional representation can be reconstructed.

To prove the applicability of this method, REMUS was deployed in the Charles River in Boston, Massachusetts and directed underneath the Massachusetts Avenue Bridge. The bridge provided obstacles that have distinct three-dimensional shapes. Using this dataset and the models and algorithms developed the main features of the bridge, the berm and pylons were reconstructed. The resulting three-dimensional grid clearly depicted the height, width, and depth of the pylons. The grid was also able to locate the rapid shoaling of the berm.

The two-dimensional representations of the bridge depict the bridge as an unnavigable area. By successfully locating these obstacles and determining the open space surrounding them, the vehicle can safely navigate underneath the bridge. Three-dimensional reconstruction of the environment extends the areas in which the vehicle can safely be deployed.

ACKNOWLEDGMENTS

Many people contributed to the success of this project and to all of them I owe a debt of gratitude. My advisor, Doug Horner, provided invaluable guidance and editing. I would also like to thank my co-advisor Roberto Cristi for his insight into this project. My fellow students at NPS are also deserving of my gratitude. The many discussions we have had enabled me to weed out many paths that would have not yielded success.

Most importantly, I would like to thank my family. Their support and patience was instrumental in enabling me to successfully complete this project.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

During the last several years there has been an explosion of interest in unmanned underwater vehicles (UUV). UUVs are being utilized to explore the ocean depths, survey and aid in salvage operations, and are even used in keeping our soldiers and sailors safe. The current crop of UUVs is utilized in areas where the main navigational hazard is the ocean bed. The next generation of UUVs will need to be able to navigate in water with a variety of obstruction: pier pylons, kelp beds, oil platforms, etc.

B. STATEMENT OF PROBLEM

UUVs are currently limited by the navigational sensors they employ. The sensors primarily map the environment into a two-dimensional representation. While this works well for unmanned ground vehicles mapping a hallway, it leads to many complications when traversing in the undersea environment. In the past UUVs have been employed in locations with a large open area. They have successfully mapped the bottom contours and other bathymetric features. However, the next generation of UUVs will need the ability to navigate among a wider variety of obstacles. For example, a UUV may be required to conduct searches in a congested harbor area and do so several times a year to measure and detect changes in the environment. This adds a new level of navigational complexity.

In order to safely navigate in an unknown environment the vehicle must be able to image the space directly in front of it. Forward Looking Sonar (FLS) can provide submerged vehicles with this capability. Current FLS map the area into two-dimensional images. During the process of compressing a three-dimensional world into a two-dimensional image information is lost. In some circumstances, that information can be critical for proper navigation.

Figure 1 is a picture of the Massachusetts Avenue Bridge in Boston MA. Figure 2 is a rough approximation of the undersea support for the bridge. There are several distinct

features. Each pylon has an associated width, height and depth. The river bottom quickly shoals on each side of the bridge. The shoaling on either side is to support the pylons.

Figure 1. Massachusetts Avenue Bridge in Boston MA, from [18]

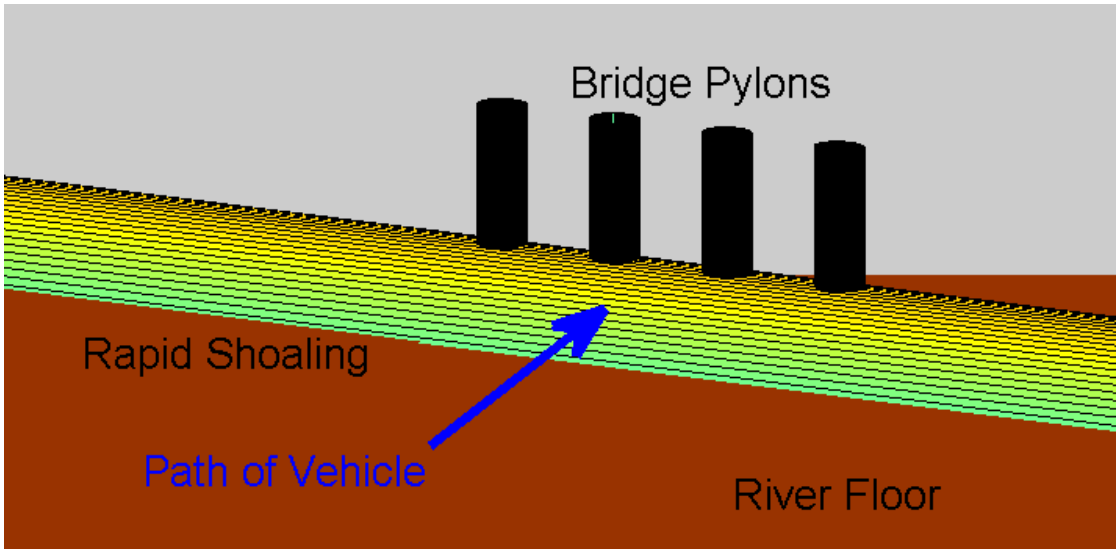

Figure 2. Submerged Bridge Pylons

Currently, UUVs are using FLS for obstacle avoidance and path planning ([1] [2]). The NPS REMUS vehicle is currently equipped with both a vertical and horizontal FLS. Figures 3 and 4 show both horizontal and vertical FLS sonar images from a UUV passing underneath the bridge.

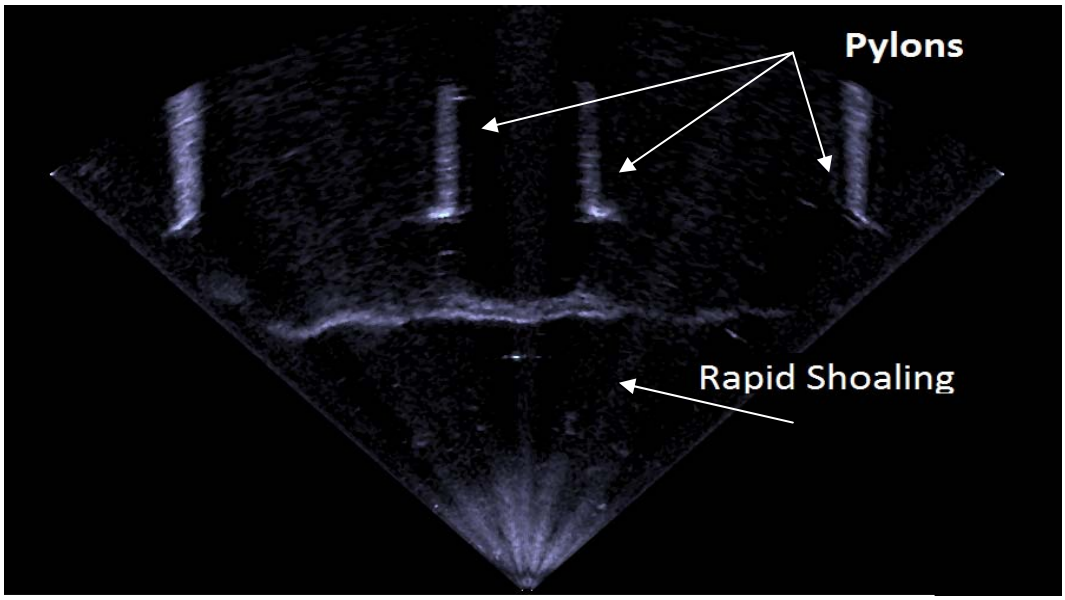

Figure 3. Horizontal FLS Image from a REMUS vehicle navigating underneath the Massachusetts Avenue Bridge

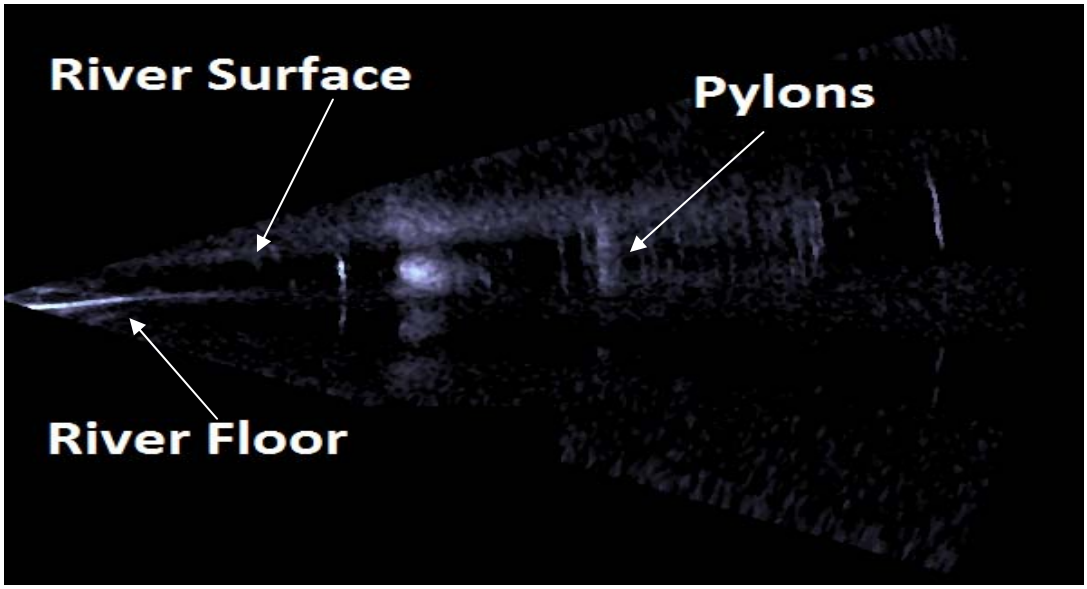

Figure 4. Vertical FLS Image from a REMUS vehicle navigating underneath the Massachusetts Avenue Bridge

The first feature of note is the quickly shoaling river bottom. According to the horizontal FLS, the change in depth of the river bottom appears as a solid wall. However, data provided by the vertical FLS shows that the vehicle can simply change its depth and safely navigate above the river bottom. The information from the horizontal image would prevent the vehicle from approaching the bridge. The pylons of the bridge present a similar problem. They are clearly visible in the horizontal image, but again show as a solid object in the vertical image. This registers with the obstacle avoidance algorithm as a rigid boundary.

The main point is that when relying either on the individual vertical and horizontal images the vehicle would not be able to find a safe path for navigation under the bridge. The bridge effectively becomes a blockade that prevents further navigation. This is a severe limitation for UUVs. In order to overcome this, a method must be developed to take the two separate sources of information and merge them into a more coherent single three-dimensional environmental model. A model that allows the vehicle to locate the opening between the pylons.

UUVs are being called upon to accomplish increasingly complex tasks, such as navigating up rivers. This requires detailed knowledge of the three-dimensional world in which the UUV is located. In order to accomplish tasks such as these, the vehicle will need the benefits provided by a three-dimensional model of the area.

C. SCOPE AND STRUCTURE OF THESIS

This thesis will look into the feasibility of reconstructing a three-dimensional environment through the use of two-dimensional horizontal and vertical forward looking sonar images. The creation of the three-dimensional environment is considered critical for the UUV to navigate in cluttered and restricted waterways. In order to create a three-dimensional model, a probabilistic model of the forward-looking sonar will be developed. There are two aspects to the model – the vehicle motion model and the sensor model. The sensor information is only as good as the estimate of the vehicle's position. The thesis will start with a description of the vehicle and a probabilistic model for estimating position and errors.

Next, the sensor model will be defined. The model will define the spatial relationship described by the sonar images. This relationship will be used to determine an object's location in global environment. This FLS model will also need to define both the probability of detection and the probability of noise attributed to each sensor.

Following the FLS model description, an occupancy grid will be developed. This three-dimensional grid will be the representation of the environment. The probabilistic model of the vehicle's position will be combined with the probabilistic and spatial models of the FLS to populate the occupancy grid. The following chapters will discuss the generation of the FLS model, the basis for occupancy grids, and will culminate with the algorithm used to create the three-dimensional model.

D. PRIOR WORK

Three-dimensional mapping has been considered for many applications. Robotic vehicles have employed various techniques to determine their environment, whether they are entering damaged buildings for disaster relief [3] or mapping the ocean floor [4]. Unmanned vehicles have also been employed with a wide range of sensors. The sensors utilized in three-dimensional mapping include cameras, laser range finders, radar, and multiple forms of sonar.

The current methods used in mapping the undersea environment are focused on producing maps after the vehicle has transited through the area. These methods typically involve a side scan sonar. Side scan sonar uses a specific array of active sonar heads to provide imagery of the ocean floor to the side of an UUV. Side Scan sonar can produce very accurate three-dimensional measurements, but will not provide images of the volume directly in front of the vehicle. With the goal of obstacle avoidance, the use of side scan sonar provides little benefits. Forward-Looking Sonars have previously been used for reactive obstacle avoidance with success ([1]& [2]). Nevertheless, in this past research the vehicle was limited to a singular plane of view, either vertical or horizontal. Little research has been conducted on combining multiple FLS into a single representation. The thesis will look into the feasibility of utilizing FLS to develop a three-dimensional map.

THIS PAGE INTENTIONALLY LEFT BLANK

II. VEHICLE DESCRIPTION

A. VEHICLE DESCRIPTION

The Remote Environmental Measuring UnitS (REMUS) is the vehicle used in this thesis. It is a multipurpose unmanned underwater vehicle (UUV) currently in use by the Center for Autonomous Vehicle Research (CAVR) at the Naval Postgraduate School (NPS). REMUS is a small man-portable UUV ([5]). REMUS specifications are listed in Table 1. This small package provides for easy deployment and recovery. Coupling its ease of deployment with a reconfigurable payload allows REMUS to be capable of executing a wide assortment of missions.

Maximum Diameter	19 cm
Maximum Length	160 cm
Weight in Air	37 kg
Trim Weight	1 kg
Maximum Operating Depth	100m
Energy	1kw-hr Lithium Ion Battery
Endurance	22 hours at 3 kts >8 hours at 5 kts
Propulsion	DC brushless motor, 3 bladed propeller
Velocity	Up to 5 kts

Table 1. REMUS Specifications, from [5]

Figure 5. NPS REMUS onboard The Cypress Sea

REMUS can be equipped with various environmental sensors including temperature and conductivity sensors, pressure sensors, fluorometer, and turbidity sensors. In addition to the various environmental sensors, it is also capable of carrying a wide range of imaging sensors, to include side scan sonar, forward looking sonar (FLS), and video cameras. Navigationally, the vehicle may choose from many options. The navigational sensors include GPS, inertial navigational system (INU), long or ultra short baseline, and Acoustic Doppler Current Profiler (ADCP). This wide range of payload options allows REMUS to be used in operations that range from scientific surveys to harbor security.

The REMUS vehicle employed by the CAVR is a modified version. The CAVR REMUS is equipped with a prototype FLS provided by BlueView Technologies. The prototype FLS will be the main sensor utilized in this thesis. This FLS will be discussed at length in the following chapters.

B. EQUATIONS OF MOTION

To take full advantage of the sensors employed, the position of the vehicle is required. Like all vehicles, REMUS position at any given time is a result of a complex set of forces. A single paper could be dedicated to the description of the forces affecting underwater vehicles. The purpose of this section is to provide the reader with a simplified overview of these forces and how they are modeled. A more detailed explanation can be found in [7].

To discuss the forces acting upon a body, a common reference frame must be defined. The standard approach to relate the position and motion in two separate frames of reference is the Newton Euler approach. The coordinate systems used in this approach can be seen in Figure 6 .

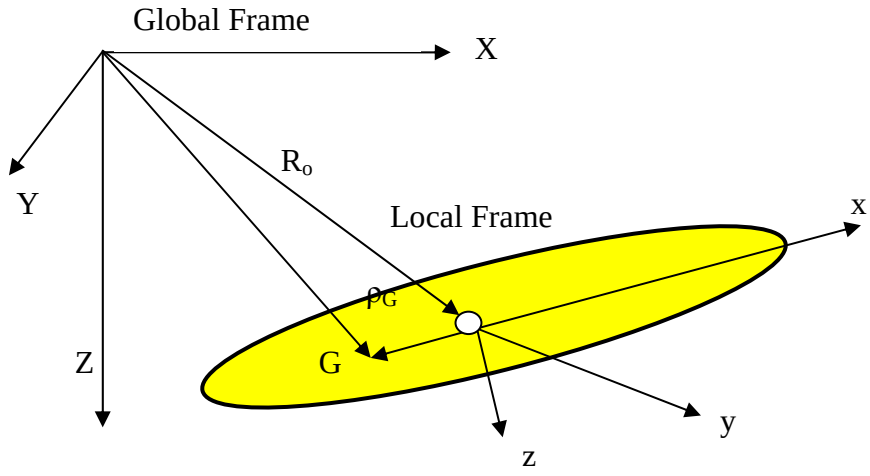

Figure 6. Local and Global Coordinate System, from [7]

The following assumptions are made for this model:

- The rotation of the earth does not appreciably contribute to the acceleration of the vehicles center of mass. Therefore, the rotation of the earth is negligible.
- The vehicle is modeled as a rigid body

- The sources of significant forces acting upon the vehicle are propulsion, hydrostatic, and hydrodynamic lift and drag. The forces can all be classified as either inertial or gravitational.

The local velocity is a vector composed of surge (u), sway (v), and heave (w). The conversion from the global velocity $\begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} \end{bmatrix}$ to the local velocity requires a transformation matrix.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = T * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad 2\text{-}1$$

T, the transformation matrix, is defined in terms of 'Euler' angles (φ,θ,ψ) and is listed in Equation 2-2.

$$T(\phi,\theta,\psi) = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin d\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix} \qquad 2\text{-}2$$

Likewise the rate of change of the 'Euler' angles can be defined in terms of the global angular velocity vector [p,q,r].

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \big/ \cos\theta & \cos\phi \big/ \cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad 2\text{-}3$$

From these equations, Healey derived six equations of motion for a rigid body. The equations of motion defined as Surge (Equation 2-4), Sway (Equation 2-5), Heave (Equation 2-6), Roll (Equation 2-7), Pitch (Equation 2-8), and Yaw (Equation 2-9). Table 2 lists the variables and definitions used in the equations of motions.

$$X_f = m\left[\dot{u}_r - v_r r + w_r q - x_G\left(q^2 + r^2\right) + y_G\left(pq - \dot{r}\right) + z_G\left(pr + \dot{q}\right)\right] + \left(W - B\right)\sin\theta \quad 2\text{-}4$$

$$Y_f = m\left[\dot{v}_r + u_r r - w_r q + x_G\left(pq + \dot{r}\right) - y_G\left(p^2 + r^2\right) + z_G\left(pr - \dot{q}\right)\right] - \left(W - B\right)\cos\theta\sin\phi \quad 2\text{-}5$$

$$Z_f = m\left[\dot{w}_r - u_r q + v_r p + x_G\left(pr + \dot{q}\right) + y_G\left(qr + \dot{q}\right) - z_G\left(p^2 + q^2\right)\right] + \left(W - B\right)\cos\theta\cos\theta \quad 2\text{-}6$$

$$\begin{aligned} K_f &= I_x\dot{p} + \left(I_z - I_y\right)qr + I_{xy}\left(pr - \dot{q}\right) - I_{yz}\left(q^2 - r^2\right) - I_{xz}\left(pq + \dot{r}\right) \\ &+ m\left[y_G\left(\dot{w} - u_r q + v_r p\right) - z_G\left(\dot{v}_r + u_r r - w_r p\right)\right] \\ &- \left(y_G W - y_b B\right)\cos\theta\cos\phi + \left(z_G W - z_B B\right)\cos\theta\sin\phi \end{aligned} \quad 2\text{-}7$$

$$\begin{aligned} M_f &= I_x\dot{q} + \left(I_z - I_x\right)pr - I_{xy}\left(qr - \dot{p}\right) + I_{yz}\left(pq - \dot{r}\right) + I_{xz}\left(p^2 + r^2\right) \\ &- m\left[x_G\left(\dot{w} - u_r q + v_r p\right) - z_G\left(\dot{u}_r - v_r r + w_r Q\right)\right] \\ &+ \left(x_G W + x_B B\right)\cos\theta\cos\phi + \left(z_G W - z_B B\right)\sin\phi \end{aligned} \quad 2\text{-}8$$

$$\begin{aligned} N_f &= I_z\dot{r} + \left(I_y - I_x\right)pq - I_{xy}\left(p^2 - q^2\right) - I_{yz}\left(pr + \dot{q}\right) + I_{xz}\left(qr - \dot{p}\right) \\ &+ m\left[x_G\left(\dot{v}_r + u_r r - w_r p\right) - y_G\left(\dot{u}_r - v_r r + w_r q\right)\right] \\ &- \left(x_G W - x_b B\right)\cos\theta\sin\phi - \left(y_G W - y_B B\right)\sin\theta \end{aligned} \quad 2\text{-}9$$

I_x, I_y, I_z	Mass moment of inertia terms
u_r, v_r, w_r	Component velocities for a rigid body fixed system with respect to the water
p, q, r	Component angular velocities for a rigid body fixed system
x_B, y_B, z_B	Positional difference between center of buoyancy and the geometric center
x_G, y_G,z_G	Positional difference between the center of gravity and the geometric center
B	Buoyancy
W	Weight
$\delta_r(t)$	Delta Rudder function

Table 2. Equation of Motion Variables

Dolbec [8] coupled the equations of motion with the hydrodynamic coefficients associated with REMUS [9] to create the final kinematics equation used to model the motion of REMUS (Equation 2-10).

$$\begin{bmatrix} m - Y_{\dot{v}_r} & 0 & 0 \\ 0 & I_{zz} - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v}_r \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_{v_r} & Y_r - mU & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r(t) \qquad 2\text{-}10$$

C. POSITION ESTIMATION

The equations of motion provide the theoretical framework for determining a vehicle's position based upon the forces acting upon it. For the equations of motion to provide any practical benefit they must be coupled with a sensor capable of measuring the acceleration and angular rates in each of the component directions. For a UUV the sensors include the following:

1. Acoustic Doppler Current Profiler (ADCP)/Doppler Velocity Log (DVL) – It provides:
   a. Water velocities – surge, sway and heave
   b. Vehicle velocities over the ground – forward, side slip and heave
   c. Vehicle Altitude
2. Depth sensor
3. Compass – Heading and heading rate
4. Global Positioning System – Latitude and Longitude position on the surface
5. Inertial Navigation Unit (INU) – Accurate angular and motion velocities

These sensors are combined together using an Extended Kalman Filter (EKF) to provide an optimal position estimation of the vehicle. The theory of the EKF, a derivative of the Kalman filter, has been discussed in depth in many publications; this chapter will serve simply as an overview of the EKF and its application to REMUS.

In a Kalman filter, the state and variance of a dynamic linear system is determined recursively. The Kalman filter is a two-step process. The first step is to predict the state

and variance of the system using the model of the system (movement update). For REMUS the equations of motion serve as this model. Upon receiving a new measurement, the difference between the measurement and the predicted state is calculated. This difference is a correction factor used to update the current state of the system (measurement update). The process is depicted in Table 3.

Movement Update	$\mu_{i+1} = \phi\mu_i + u_w$ $\Sigma_{i+1} = \phi\Sigma_i\phi' + Q$
Measurement Update	$K_{i+1} = \frac{\Sigma_{i+1}H'}{H\Sigma_{i+1}H' + R}$ $\mu_{i+1\|Z} = \mu_{i+1} + K\left(Z - \mu_v - H\mu\right)$ $\Sigma_{i+1\|Z} = \left(I - KH\right)\Sigma_{i+1}$
Variable Definition	(μ, Σ): mean and covariance of the system state (μ_v, R): mean and covariance of the measurement noise (μ_w, Q): mean and covariance of the movement noise H: measurement matrix K: Kalman Gain Z: measurement ϕ: movement matrix

Table 3. Kalman Filter Equations, after [10]

The method listed in Table 3 will only work for systems with a linear model. However, modeling the position of REMUS is a non-linear process. Since REMUS does not measure position directly, but deals with the estimation of position from both angular rates and accelerations the equations of motion are non-linear in nature. Because of the non-linearity an EKF is required. In an EKF, the nonlinear components are linearized by an approximation function (normally a Taylor series approximation). In an EKF the following substitutions are made:

- (μ, Σ) is replaced with $f(\mu, \Sigma)$
- Z is replaced with h(Z)

The most accurate sensor for measuring the global position of the REMUS was GPS. This is only available when the vehicle is on the surface. Once underwater, position estimation errors grow more rapidly over time. For the UUV to navigate accurately for extended periods of time underwater, it requires a high-grade inertial navigation system. The REMUS UUV had an integrated navigation suite that combines together the IMU, GPS and ADCP/DVL. It is the SEA DeViL T24, a navigation suite from Kearfott that advertises a position accuracy of 0.5% total distance traveled. The stated manufacture specifications for REMUS' IMU are listed in Table 4. Alameda provides a detailed explanation of the Sea DeViL architecture [11].

SEA *DeViL* PERFORMANCE*			
Surface Ship	**KN-6051**	**KN-6052**	**KN-6053**
Position Accuracy			
• GPS/DVL**	10 m, CEP	10 m, CEP	10 m, CEP
• DVL (Water-Track Mode)****	10 nm/8hrs, TRMS	2 nm/8hrs, TRMS	1 nm/8hrs, TRMS
Heading Accuracy			
• GPS/DVL**	5.0 mils, rms	<1.5 mils, rms	<1.0 mils, rms
• DVL	5 mils* secant λ, rms	1.0 mils* secant λ, rms	0.5 mils * secant λ, rms
Velocity Accuracy			
• GPS/DVL**	0.05 m/sec	0.05 m/sec	0.05 m/sec
• DVL (Water-Track Mode)	0.5 m/sec, rms	0.35 m/sec, rms	0.3 m/sec, rms
Roll/Pitch Accuracy	0.5 mils, rms	0.5 mils, rms	0.5 mils, rms
Underwater Vehicle***	**KN-6051**	**KN-6052**	**KN-6053**
Position Accuracy ****	0.5% DT, CEPR	0.2% DT, CEPR	0.05% DT, CEPR
Heading Accuracy	5 mils*, rms	1.5 mils*, rms	1.0 mils *, rms
Roll/Pitch Accuracy	0.5 mils, rms	0.5 mils, rms	0.5 mils, rms

Table 4. Sea DeViL Performance From Kearfott Systems, from [12]

During this experiment, the vehicle did not receive any position measurements after the initial dive. The position and the errors associated with it will be entirely attributed to the accuracy of the Sea DeViL. Due to the short relative distance traveled during experiment, the errors in position are minimal. The vehicle traveled approximately 450 meters to reach the bridge. This equates to a possible positional error

of 2.3 meters. The total distance traveled at the beginning of the last pass underneath the bridge was 1350 meters, yielding a maximum expected error of 6.5 meters. For this reason, the position and orientation will be assumed to be relatively accurate.

The position of the vehicle (or mobile sensor) and its orientation play a critical role in mapping the output of the sensor to the global space. All of the information provided by the sensor is referenced to the vehicles location and attitude. Any error in the vehicles location is propagated through each of the models. In other words, errors in the vehicle's position or orientation will cause objects to be plotted in the wrong location, invalidating the entire method.

The next chapter will discuss the model of the FLS. The FLS model will then be combined together with the vehicle model in order to take the sonar images and reconstruct a three-dimensional occupancy grid that represents underwater objects.

THIS PAGE INTENTIONALLY LEFT BLANK

III. FORWARD LOOKING SONAR

A. INTRODUCTION

A sonar uses the distribution and reflection of a sound wave to determine the range and bearing to submerged objects. Active sonar systems use a transducer to create their own source of sound in the water (called a ping) and then listen for its return. In its most basic form, active sonar determines the distance to a given target by measuring the time delay from the production of a sound until the return reaches the transducer. In order to determine the distance to a target, all that is required is the total travel time and the speed of sound through the water. This yields Equation 3-1.

$$Range = \frac{\text{time delay} * \text{Speed of sound}}{2} \qquad 3\text{-}1$$

The accuracy of the range is then dependent upon the ability of the system to accurately measure the time delay and the accuracy of the speed of sound. Ideally, the speed of sound would be constant throughout the water, however, in practice this is not the case. The approximate speed of sound is calculated onboard the REMUS vehicle using water temperature and salinity. Normally, this is about 1500 meters per second.

Ranging information alone is not sufficient to geo-locate an object. To accurately determine the location of the object it's bearing must also be determined. To measure the bearing to an object, a transducer will only project or listen to sounds along a particular bearing (also called a beam). Thus by only listening to sounds along a particular bearing and measuring the time delay of the sounds return, both the bearing and the range to an object can be determined. The ability of the transducer to listen along a particular bearing or range of bearings is called its directivity. A transducer with a high directivity will be able to isolate sounds to a small range of bearings. Transducers with low directivity have larger bearing resolutions.

For the sonar to determine the range and the bearing to an object it must be able to distinguish the return from the background noise. There are numerous sources of noise for any sonar system. In the underwater environment, multiple sources of sound exist.

Everything from biologics to merchant traffic emits sounds across all frequencies. Wave action and rain also contribute significantly to the amount of background noise a sonar system must overcome. The overall ability of the sonar system to determine detection is based on the ratio of return signal received at the transducer to the amount of general noise received at the transducer. This is the sonar's signal to noise ratio (SNR).

Due to the larger range of values of sound energy in the water, SNR is generally defined in terms of decibels (dB). The general equation for SNR in an active sonar system is shown in Equation 3-2.

$$SNR = SL - 2TL + TS - NL \qquad 3\text{-}2$$

The amount of signal returned to the transducer is dependent upon the amount of sound initially projected into the water, the source level (SL). After leaving the transducer, the sound expands and travels through the medium towards the object. While traveling, the intensity of the sound is diminished via attenuation and expansion. This loss of intensity is called the transmission loss (TL). The total amount of sound received at the object is the source level minus the transmission loss.

Once the sound reaches an object in the water, it is then reflected off the object. The amount of sound reflected and directed back towards the original source of the sound is called the target strength (TS). Many factors affect the amount of sound reflected off the object. TS is dependent upon the angle of incidence, the absorption properties of the object, the size and shape of the object, etc. Once the sound is reflected, it travels back towards the source. As it travels, it undergoes the same amount of transmission loss. So the total pressure received at the transducer from original signal is

$$\text{Received Signal} = SL - 2\,TL + TS \qquad 3\text{-}3$$

However, this does not take into account the amount of noise the transducer receives. Letting the noise level (NL) be defined as the amount of noise received by the transducer, the SNR equation becomes

$$SNR = SL - 2TL + TS - NL \qquad 3\text{-}4$$

B. BLAZED SONAR ARRAYS

Blazed array sonar is a method to create a highly directive signal in a small package. Blazed array sonar systems utilize a principle similar to echelette diffraction gratings in optics to create a multitude of beams from a single sound source [13]. The blazed sonar array maps the frequency of a given broadband pulse to the angular spatial domain. The principle is similar to that of a prism. A broad spectrum signal (white light) is processed through the stave (the prism) and individual frequencies are projected to independent spatial locations (the rainbow). Each frequency band creates a unique beam. The shape and size of the beam is dependent on the frequency band and the shape of the stave projecting the sound. A plot of the magnitude of the projected sound versus bearing for a typical blazed array sonar is provided in Figure 7. The lower larger beams were created by the lower frequencies.

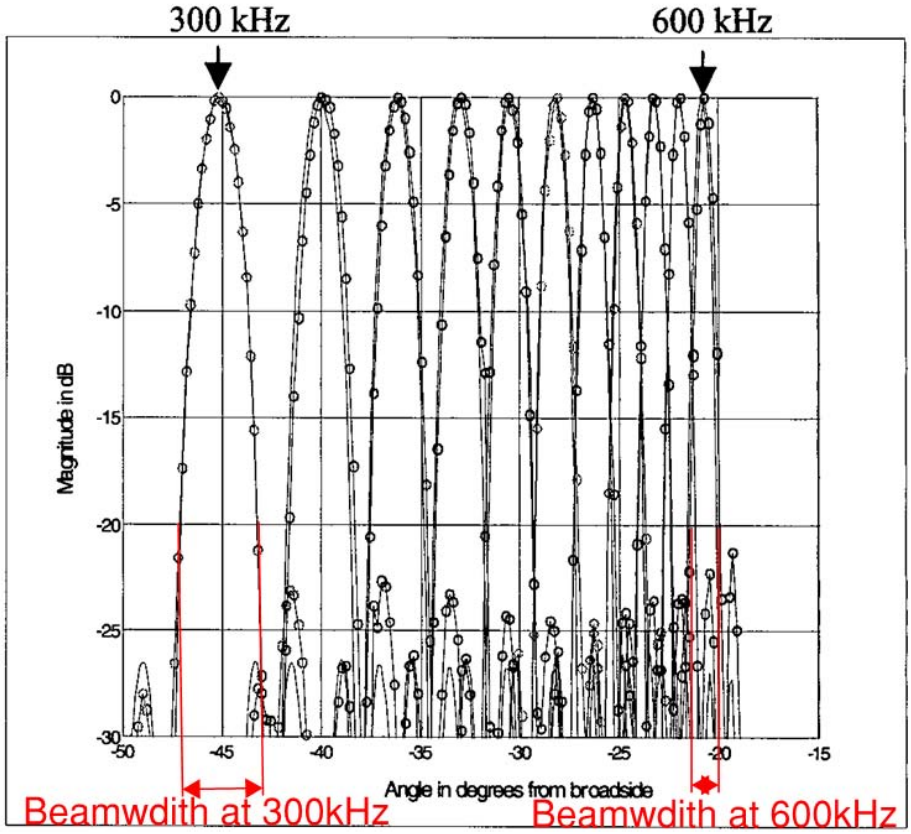

Figure 7. Composite Blazed array beam patterns for frequencies between 300 kHz and 600 kHz, from ([15])

Once the sound has been transmitted into the water from the blazed array, it interacts with its environment just like any other active sonar. The main difference is that the frequencies are separated into individual beams. As all the frequencies return to the transducer, they are recombined into a single signal. The recombination of the signals occurs in reverse order of the creation. Each beam will only receive a particular band of frequencies. This maintains the original frequency to angular spatial mapping.

Since each frequency band correlates to a specific angle in the spatial domain, the individual frequencies can be modeled as a single hydrophone with a high directivity. Maintaining the frequency to angular spatial relationship creates a simulated array of hydrophones. This simulated array is what creates the detailed representation of the volume ensonified.

C. BLUEVIEW P450-15E

The BlueView P450-15E is a blazed array system with a center frequency of 450 kHz. The P450-15E specifications are listed in Table 5.

P450E-15 Sonar

Max Range:	450ft
Update Rate:	Up to 10Hz
Swath Width	45º
Beam Width	1º x 15º
Electrical	
Power	12-48 volts @ 10 watts (with 50ft cable)
Communications	Ethernet, 10base-T or 100base-T
Mechanical	
Depth Rating	1000ft
Weight in air	5.75 lbs
Weight in fresh water	1.4 lbs
Dimensions (Max)	9.5" x 7" x 4"
Acoustic	
Operating Frequency	450kHz

Table 5. BlueView P450-15E Specifications (from BlueView Inc., 2008)

The P450-15E produces image files in both Cartesian and Polar coordinates (see Figure 8). The resolutions available according to image type are listed in Table 6. While the P450-15E is capable of higher range resolutions, it is constrained by the size of the image available. Due to the nature of the blazed array sonar, the polar coordinate system is the natural coordinate system. In order to transfer the information from the polar coordinate system to the Cartesian image, pixels with changing resolutions were required. In the polar image the resolution per pixel is constant throughout the entire image. For these reasons, polar images where utilized in this thesis.

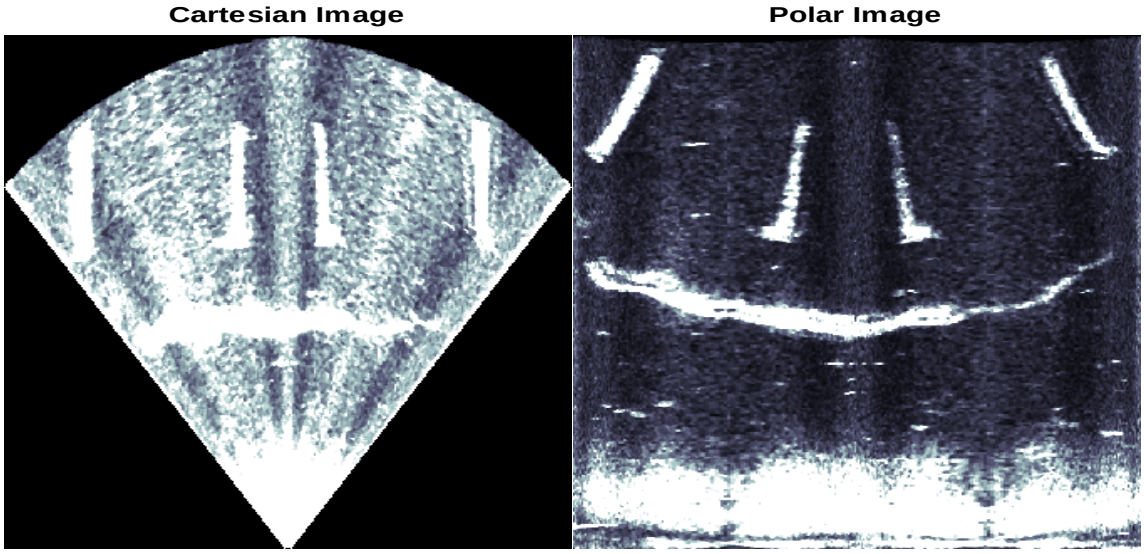

Figure 8. Cartesian and Polar images from P450-15E sonar

	BlueView Specification	Cartesian	Polar
Image Size (in pixels)		387 x 571	461 x 2048
Range Resolution	2 inches	Appoximately 12” x 6”	$\frac{90m}{461 \text{ pixels}} = 7.67 \, {}^{in}/_{pixel}$
Bearing Resolution	1 deg	varies based on the location of the pixel	$\frac{90^o}{2048 \text{ pixels}} = .044 \, {}^{\deg}/_{pixel}$

Table 6. Comparison of Horizontal FLS Image Resolutions

Both image files provide an angular resolution greater than the manufacture specifications. This anomaly is attributed to the varying beamwidths of the blazed array sonar. The larger bearing resolution listed by the manufacture corresponds to the larger beamwidth of the lower frequency beam. However, the resolution of the polar image appears to be based upon the beamwidth of the highest frequency beam.

The image created by a P450-15E is a two-dimensional representation of a three-dimensional volume. A depiction of the actual volume ensonified is depicted in Figure 9. A single stave of the P450-15E has an approximate 23-degree field of view in the image plane. The NPS REMUS AUV has four staves in the horizontal plane for an approximate 90-degree total field of view.

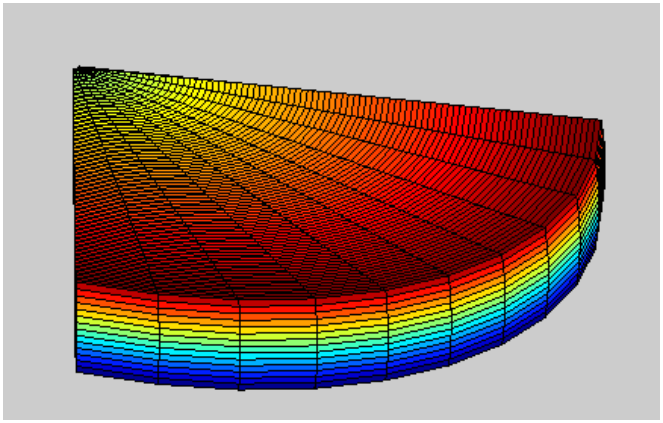

Figure 9. Volume ensonified by the horizontal P450-15E

The ensonified volume covers a fifteen degree spread in the vertical plane. The sonar is unable to distinguish the difference between an object at different elevations. For example the object highlighted in Figure 10, could be located anywhere within the fifteen-degree spread. This is the ambiguity of the sonar in the vertical plane. The object shown in Figure 10 is approximately thirty meters in front of the vehicle. Based on its range and the bearing ambiguity of the FLS, the object can be located within +/- 4 meters in altitude. The P450-15E provides a range resolution of 2 inches and a bearing resolution in the image plane of 1 degree. This results in the ensonified volume being divided into cube like shapes with dimensions of 2 inches by 1 degree by 15 degrees (see Figure 11).

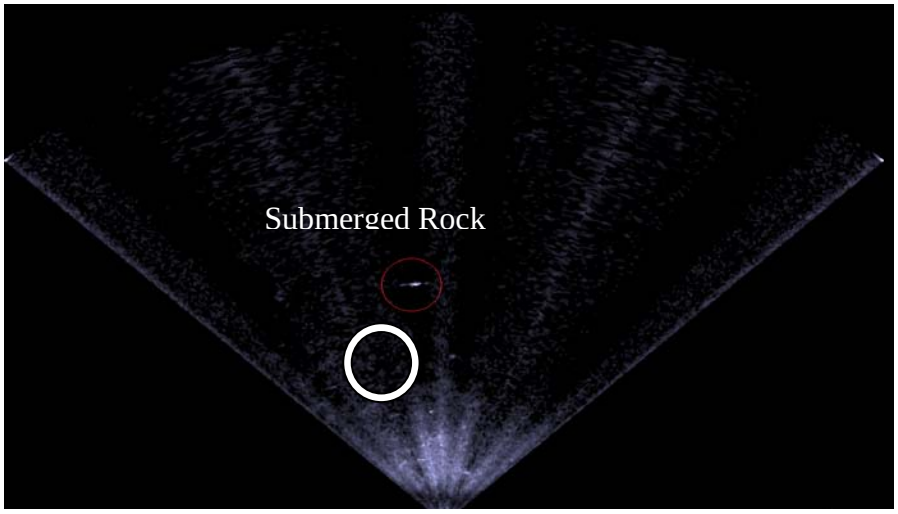

Figure 10. Submerged Rock in a Horizontal FLS Image

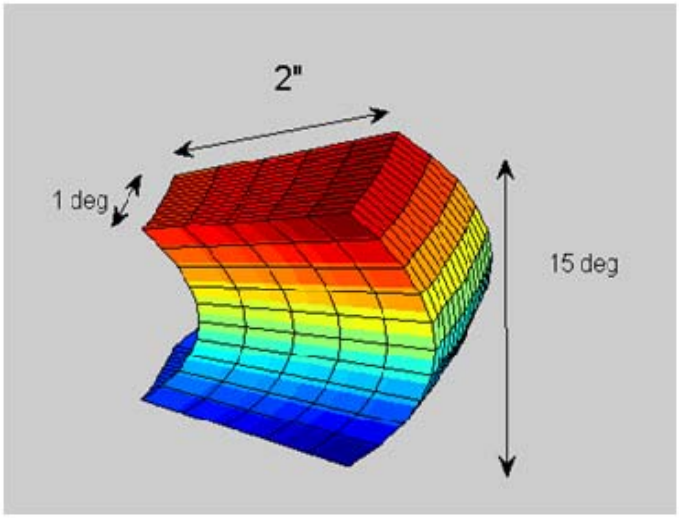

Figure 11. Volume segments created by P450-15E

The vertical array is composed of a single stave mounted perpendicular to the horizontal array. Mounting the array in this fashion provides angular resolution in the vertical direction. Due to the rotation of the array, the ambiguity of the sonar shifts to the horizontal direction. All vertical sonar images will have fifteen degrees of ambiguity in the horizontal direction. The vertical sonar will be utilized to provide the location of the ocean floor and the height of the objects encountered. A typical vertical sonar image is displayed in Figure 12.

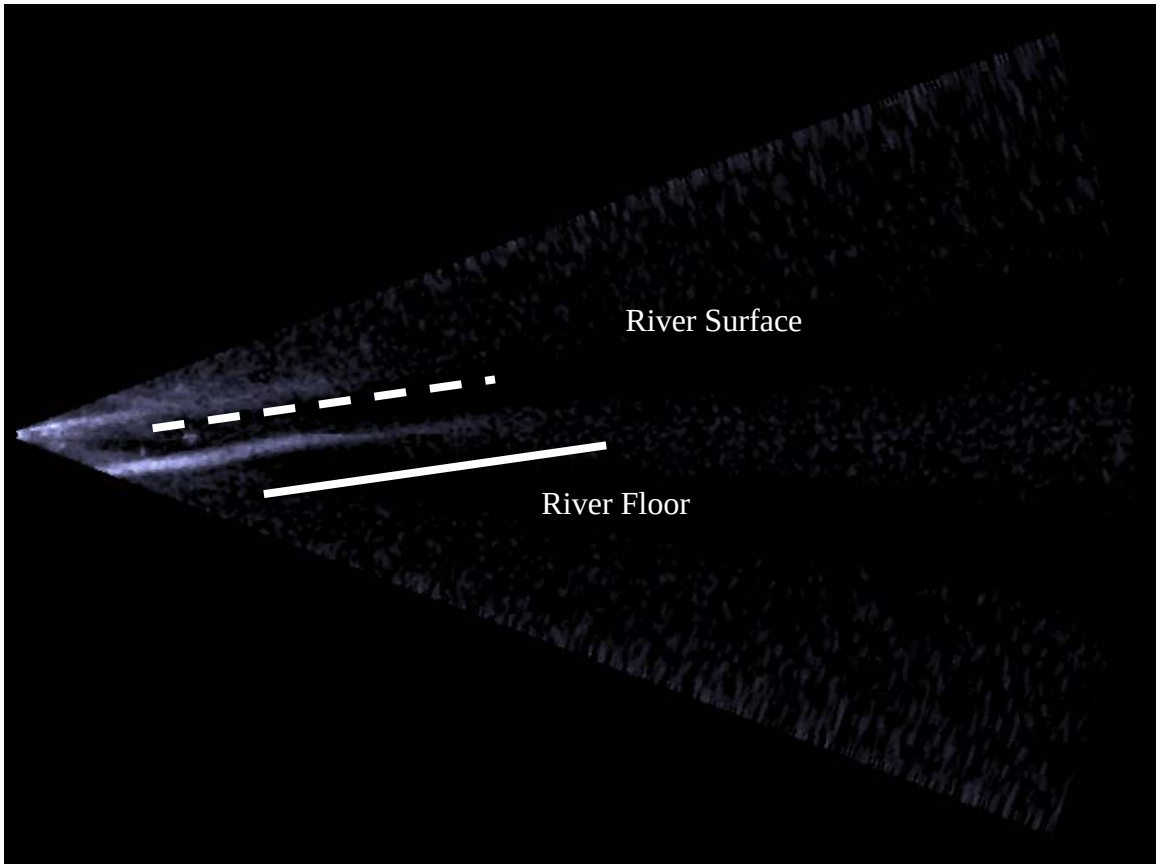

Figure 12. Vertical Sonar Image depicting the river surface and floor

By combining the separate fields of view, the two sonars can effectively create a volume with resolution in all three planes. The volume ensonified by the combination of sonars is depicted in Figure 13. Of particular importance is the fifteen-degree by fifteen-degree area where the two arrays overlap. This overlapping of the arrays divides that volume of water into sections that are two inches in depth (vertically and horizontally) by one degree in vertical width and one degree in horizontal width (Figure 14).

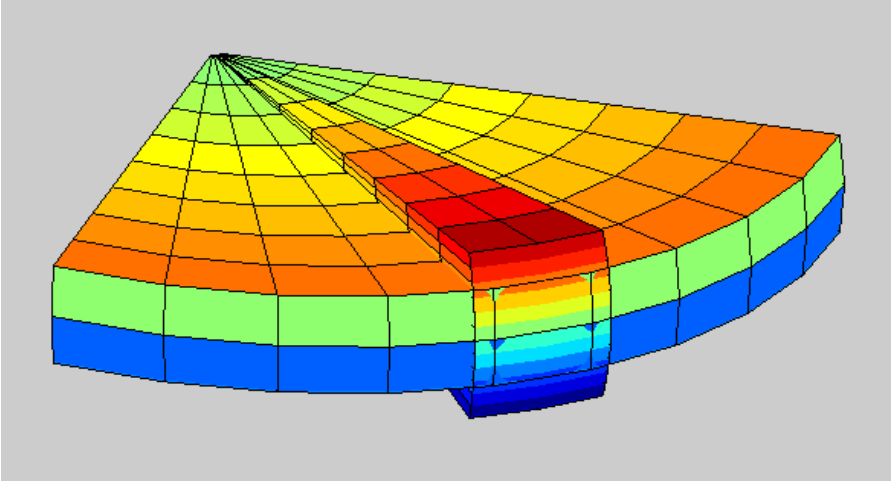

Figure 13. Volume ensonified by the combination of Vertical and Horizontal Arrays

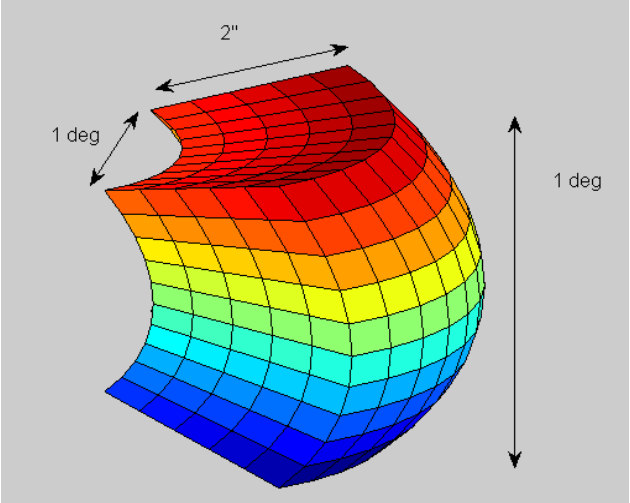

Figure 14. Volume segments created by the overlapping sonar

The vertical and horizontal sonar are independent sonar systems. Information is not shared between them. Separate two-dimensional representations are created by each system. The horizontal system provides resolution in the horizontal plane, but is unable to provide resolution in the vertical plane. The vertical system, though unable to provide resolution in the horizontal, provides resolution in the vertical. In order to create a three-dimensional model the two separate data streams need to be combined. The following chapters will discuss the use of occupancy grids to combine the separate systems into a single three-dimensional model.

D. FLS PROBABILITY MODELS

Knowing that the purpose of the FLS sonar is to populate an occupancy grid necessitates the development of probabilistic models of the FLS. For a sensor to be used in an occupancy grid, two separate probabilistic models need to be generated. One model describes the probability of receiving a value when the return can be attributed solely to noise. The second model describes the probability of receiving a particular value when the sonar return can be attributed to a physical object. This section will focus on the development of the probability models and the details of an occupancy grid will be discussed in the next chapter.

1. Horizontal FLS Noise Model

There are many possible sources of noise for the sonar system. Like all electrical systems, a certain amount of noise can be attributed to the various types of electrical noise: power supply fluctuation, thermal noise, etc. A secondary source of noise is acoustic. Acoustic noise includes weather, biological sources, or other vehicles in the area. Acoustic Models have been created for the various types of noise, but these models do not take into account the method of beamforming used in blazed array sonar. The final category of noise can be attributed to REMUS itself. REMUS utilizes several other sensors that also project noise into the water column. These sensors include but are not limited to the acoustic modem, ADCP and Side Scan sonar.

Instead of designing and incorporating theoretical models for each of the noise sources an experimental approach was taken to create a comprehensive noise model. Within the dataset used, there are approximately 300 images for the horizontal FLS that show little to no objects in the field of view. These images were acquired with the vehicle in motion and in the same acoustic environment. By using images taken in the same acoustic environment, the noise can be classified as a single distribution.

The expected distribution of the noise will be a combination of multiple noise sources. Because the resulting distribution of the noise is unknown, a histogram was used to observe the distribution. By using a histogram, an initial assumption of the

distribution does not need to be made. When creating a histogram, the bins size and spacing can play an important role is determining the distribution.

The number of bins will be determined by the range of values expected. Each image in the data set is a sixteen bit image, so each pixel has a potential range of value from zero to 65535. However, prior experience has shown that the maximum expected value of noise is much lower. In determining the bin size and spacing a subset of 50 images were analyzed. The combined histogram of each of the images was plotted. In these fifty images, the maximum value of noise was 3958. Based on the maximum noise value and considering the computer storage limitations the noise histogram contained 25 bins with a spacing of 160 (with the exception of the first and last bin). The first bin contained values from zero to 80; the last bin contained values from 3920 to 65536.

With the bin size determined, the images were analyzed in more detail. Due to the nature of the FLS, the noise distribution varied based on the range and bearing of the pixel. As shown in Figure 15 the distribution cannot be readily classified. The distribution of noise in the larger ranges appears to be a Gaussian distribution, where the distribution at close range is a combination of distributions. This more complex probability distribution function may be attributed to the near field effect of active sonar. To take into account the variability of the distributions across the entire image, a distribution was defined for each pixel in the image.

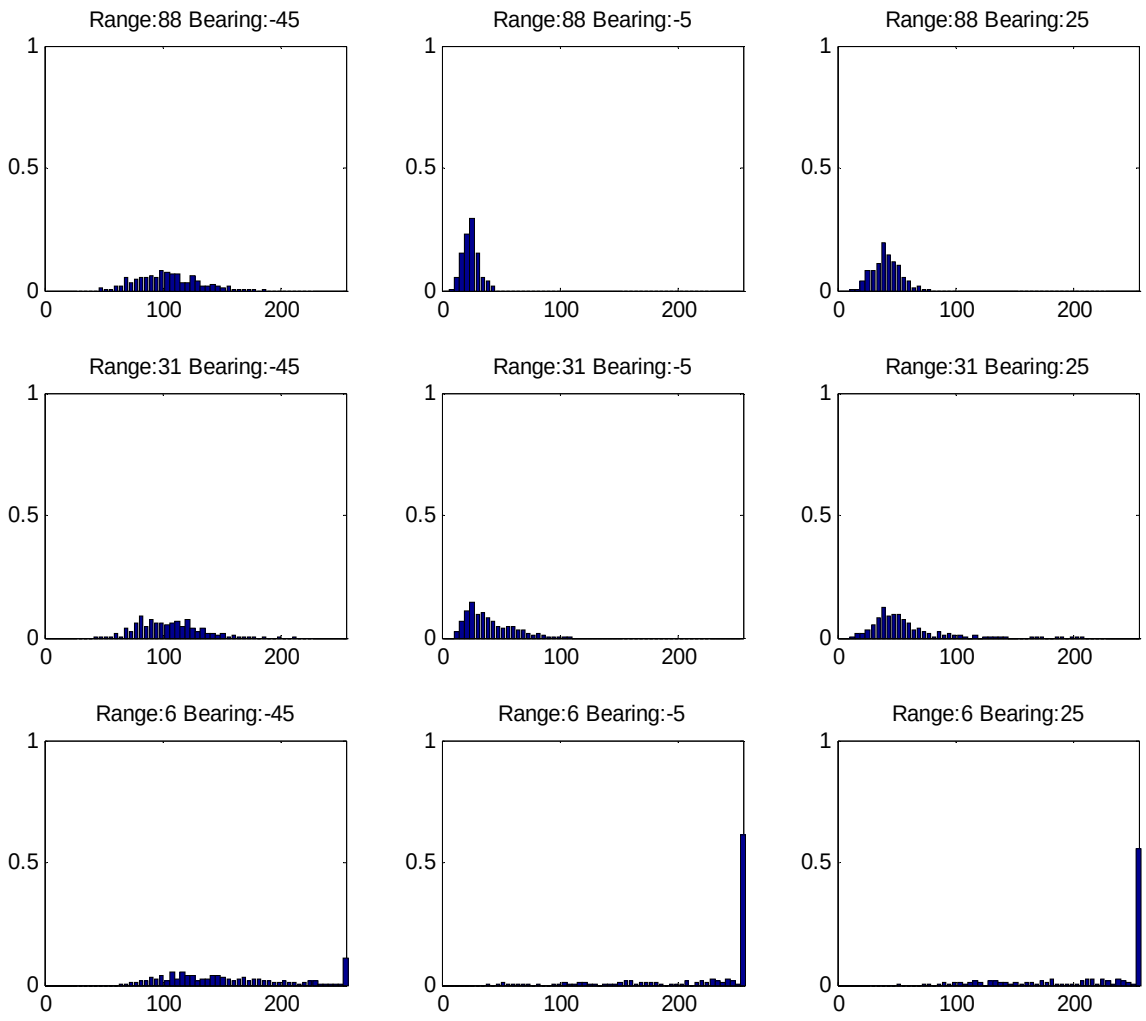

Figure 15. Noise Distribution Changes over Range (meters) and Bearing (degrees)

2. Horizontal FLS Detection Probabilities

The other distribution required for the occupancy grid is the probability of receiving a particular value given that the value is attributed to a return from an object. Ideall, this model would be based on a theoretical approach. Theoretical models exist for determining sound propagation in the water. Using the models, the intensity level of sound can be predicted at any point in the water column. However, the algorithms utilized by BlueView in mapping the broadband signal received at the FLS stave into an

image are proprietary. Without knowledge of the exact algorithms involved in generating the FLS images, an experimental approximation must be made.

A single experiment has not been conducted to determine the distribution of expected received values at every pixel in the image. To determine the distribution of received FLS image values when an object is present a subset of the Charles River dataset was analyzed. Unfortunately, every image in the Charles River dataset is subject to all the noise sources previously discussed. The noise distribution must be separated from the signal distribution.

The subset of images used for this portion of the experiment was chosen due to the large area that the objects filled in the field of view. In a similar fashion to the noise distribution, the histogram of each pixel was determined. The following process was used to separate the detection distribution from the noise distribution, and is illustrated in Figure 16 . This procedure is done for each pixel in the image.

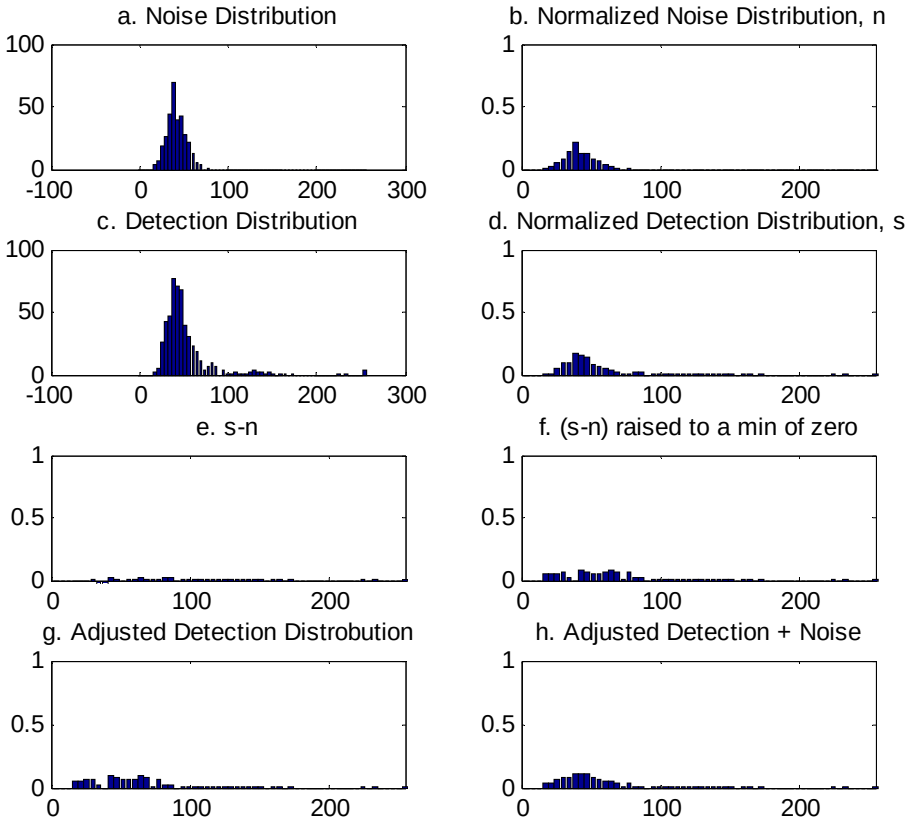

Figure 16. Determining the Probability of Detection for pixel (50, 400) of the Horizontal FLS

1. Determine $P[\text{pixel value} \mid \text{no Object}]$ (Figure 16a). This is a histogram of the raw data retrieved from the dataset.
2. Normalize $P[\text{pixel value} \mid \text{no Object}]$ (Figure 16b). In this step the histogram is normalized so that the sum of all of the bins equal one.
3. Determine $P[\text{pixel value}]$ (Figure 16c). The raw histogram of the dataset that contains images with a combination of objects.
4. Determine $P[\text{pixel value} \mid \text{Object}]$. The probability of receiving any pixel value is a summation of multiple probabilities (Equation 3-5).

$$P[\text{pixel value}] = P[\text{pixel value} \mid Object]P[Object] + P[\text{pixel value} \mid \text{no Object}]P[\text{no Object}] \quad 3\text{-}5$$

Rearranging Equation 3-5 yields Equation 3-6.

$$P[\text{pixel value} \mid \text{Object}]\frac{P[\text{Object}]}{P[\text{no Object}]} = \frac{1}{P[\text{no Object}]}P[\text{pixel value}] - P[\text{pixel value} \mid \text{no Object}] \quad 3\text{-}6$$

$P[\text{no Object}]$ was estimated by the ratio of the sum of histogram bins of $P[\text{pixel values}]$ that have a corresponding non zero bin in $P[\text{pixel values|no Object}]$. Figure 16d shows $\frac{1}{P[\text{no Object}]}P[\text{pixel values}]$.

5. $P[\text{pixel value|Object}]\frac{P[Object]}{P[\text{no Object}]}$ is shown in Figure 16e.
6. Ensure that all values of $P[\text{pixel value|Object}]\frac{P[Object]}{P[\text{no Object}]}$ remain positive (Figure 16f). If any value in the histogram generated in step five is negative, raise all values of the histogram by the maximum negative value. This ensures that all probability values are non-negative.
7. Determine $P[\text{pixel values} \mid \text{Object}]$ (Figure 16g). Assume $\frac{P[Object]}{P[\text{no Object}]}$ is a constant and normalize $P[\text{pixel value|Object}]\frac{P[Object]}{P[\text{no Object}]}$. This provides $P[\text{pixel values} \mid \text{Object}]$, the final distribution used to model the probabilities of receiving a value given that an object is present.
8. Ensure $P[\text{pixel values} \mid \text{no Object}] + P[\text{pixel values} \mid \text{Object}]$ maintains the relationship of the original distribution (Figure 16h). To verify that the separate distributions are accurate, they are summed and normalized. The

new normalized combination distribution is compared to the original normalized distribution. As can be seen in Figure 16d and h, the resulting shape of the distribution is similar to the original.

3. Vertical FLS Model

The Vertical FLS is comprised of staves identical to those used in the horizontal FLS. The difference between the vertical and horizontal FLS is the number of staves used and the orientation in which they are mounted. The vertical FLS covers 45 degrees in the field of view (2 staves) and the horizontal FLS covers 90 degrees (4 staves). The central 45 degrees of the horizontal sensor distributions should be identical to the distribution of the vertical FLS.

4. Verification of the Probability Models

The models are created on a pixel-by-pixel basis; therefore the horizontal model is comprised of almost two million individual probability functions (461x2048x2). With that number of distributions it was impractical to inspect each distribution. To verify that the relationship between the noise and detection distributions were valid, a sample Bayesian probability image was created for each sensor.

$$\text{Bayesian Image} = \frac{P\left(I \mid \text{ all pixels are due to detections}\right)}{P\left(I \mid \text{ all pixels are due to detections}\right) + P\left(I \mid \text{ all pixels are due to noise}\right)} \qquad 3\text{-}7$$

If the distributions are correct then the Bayesian image will show a high intensity value for any obstacle in the image and will show a very low intensity in areas due to noise. Figures 17 and 18 show the original FLS image with the resulting Bayesian image. As can be seen in both figures, the amount of noise is drastically reduced and the objects detected are greatly enhanced. The clarity of the images proves the validity of both the noise and detection distributions. As can be seen Figure 17, further future work can be done to remove the noise in the lower portions of the image.

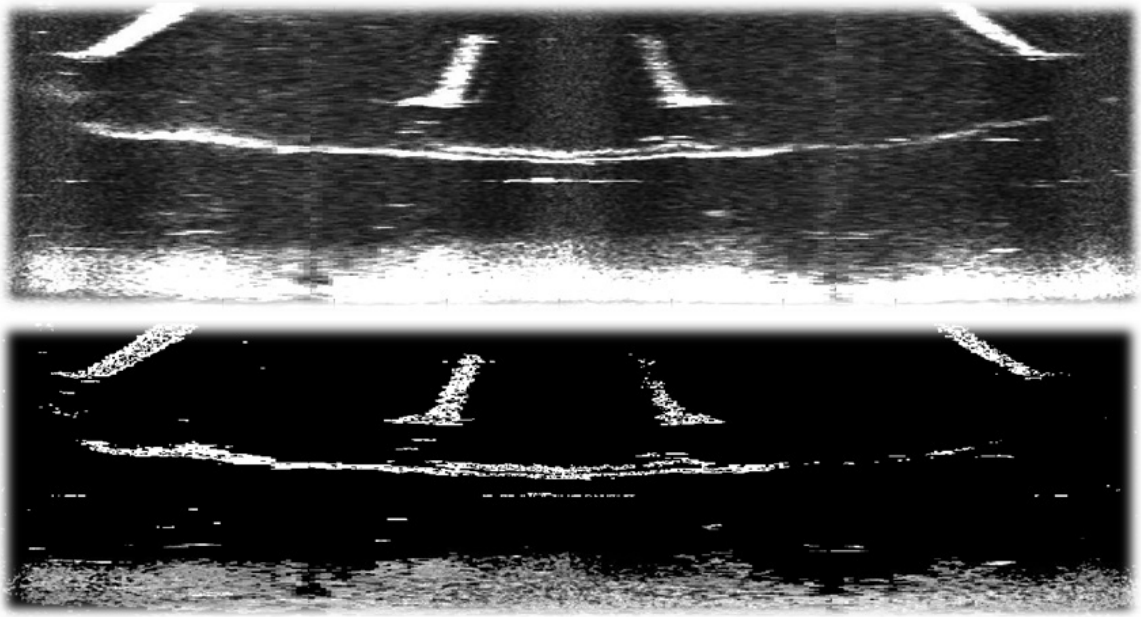

Figure 17. Horizontal FLS Image (upper) and corresponding Bayesian Image (lower)

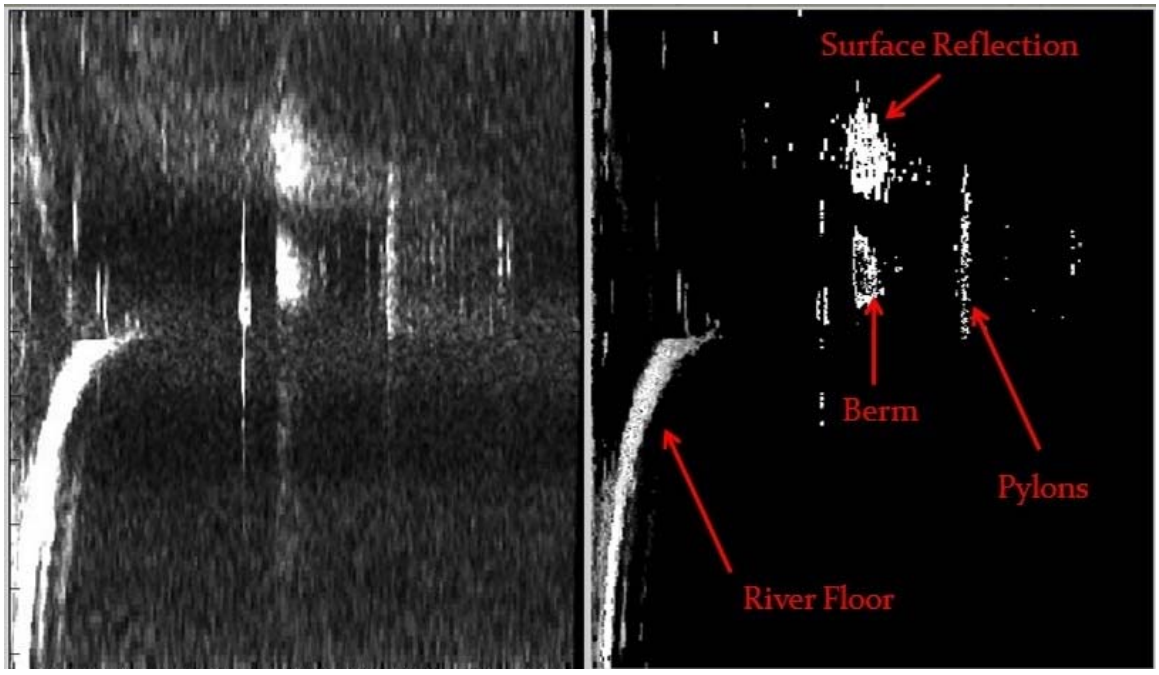

Figure 18. Vertical FLS Image (left) and corresponding Bayesian Image (right)
Both images are in polar coordinates

E. GEO-LOCATING IMAGE DATA

Up to this point, all of the information provided by the FLS has been referenced to the vehicle. To be truly useful in a mapping algorithm, the FLS data will need to be mapped into the global space. By combining the vehicle position estimation model with the FLS model, each data point in a sonar image can be geo-located. This section will cover the equations necessary to couple the models and convert sonar images into to map coordinates.

1. Development of the Transformation Equations

The height of the image corresponds to a range in front of the vehicle (0 to 90m) and the width of the image corresponds to the bearing from the vehicle. The horizontal images are 461 x 2048 pixels. The vertical images are 461 x 1024 pixels. Both of these are using polar coordinates In the horizontal images, with the four staves, the bearing is the spread in the horizontal plane of the vehicle and covers -45^o to 45^o. The vertical images cover from -22^o to 22^o in the vertical plane of the vehicle (There are two staves and the coverage is a total of 45 degrees). Equation 3-8 shows the conversion from pixel indexes into the range and bearing from the vehicle.

Horizontal Image	Vertical Image
$r_p = i * \frac{\text{Max Range}}{\text{Image Height}}$	$r_p = i * \frac{\text{Max Range}}{\text{Image Height}}$
$\theta_p = -45^o + j * \frac{90^o}{\text{Image Width}}$	$\theta_p = 0$
$\psi_p = 0$	$\psi_p = -12^o + j * \frac{24^o}{\text{Image Width}}$

3-8

where

r_p : Range in front of the vehicle

θ_p : rotation about the z axis in the XY plane

ψ_p : rotation about the x axis in the YZ plane

(i, j) : Pixel location (height, width)

To fully describe the volume ensonified, Equation 3-8 needs to be revised to include the ambiguity of the FLS. At this point, a decision needs to be made to determine

the resolution associated with the ambiguity of the FLS. Since the overall goal of the system is to combine both sensors into a single coherent sensor model, the resolution of the ambiguity is chosen to match the resolution of the opposing sensor. This provides a one-degree spacing in the ambiguity of the sonar images and causes θ_p and φ_p to become vector quantities.

Horizontal Image	Vertical Image
$r_p = i * \frac{90m}{461}$	$r_p = i * \frac{90m}{461}$
$\theta_p = -45^o + j * \frac{90^o}{2048}$	$\theta_p = \{-7, -6, ...7\}$
$\psi_p = \{-7, -6, ...7\}$	$\psi_p = -12^o + j * \frac{24^o}{1024}$

3-9

Combining the knowledge of the vehicles position and orientation with Equation3-9, allows for each pixel in the image space to be geo-located. Define the sensor's location as (x_o, y_o, z_o) with an orientation of (θ_v, φ_v, ψ_v). Due to the small amount of roll associated with the vehicle it will not be accounted for in the transformation.

(x_o, y_o, z_o): Vehicle postion in local reference frame
θ_v : Vehicle Heading (after conversion into the local frame)
φ_v: Vehicle Pitch
ψ_v: Vehicle Roll

3-10

The data provided by a single pixel can then be geo-located using Equation 3-11.

$$x_p = x_v + r_p \cos(\theta_v + \theta_p)\cos(\phi_v + \phi_p)$$
$$y_p = y_v + r_p \sin(\theta_v + \theta_p)\cos(\phi_v + \phi_p)$$
$$z_p = z_v + r_p \sin(\phi_v + \phi_p)$$

Note: $\phi_p, \theta_p, x_p, y_p, z_p$ are vectors

3-11

Equation 3-11 will map any data stored in the image space into the global space, whether the data is the original sonar image or a processed version. This fact will be utilized in the next section during the creation of the occupancy grids. Since the

occupancy grids use a probabilistic approach, each image will be transformed from the original image into the probabilistic image. The probabilities assigned to each pixel will then be mapped into the occupancy grid. The next chapter will utilize both the sonar and vehicle models to generate the occupancy grid.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. OCCUPANCY GRIDS

A. MAP CHOICE

The main goal of providing the UUV with a three-dimensional map is safe and accurate navigation. A safe route is one in which the vehicle has a low probability of collision. In other words, the vehicle is able to locate and avoid the obstacles in its path. The accurate route is one in which the vehicle is able to successfully move from a starting position and reach its end position. This thesis will not delve into the intricacies of route planning, but route planning defines the type of map required for safe navigation and thus must be considered. These goals provide the requirements for the map. First, the vehicle needs to know location of obstacles. Second, the vehicle needs to know the location of free space.

The most intuitive method to create the map is feature recognition. In feature recognition every object detected by the sensor is classified. The object's shape, dimensions and location are then stored in a database. Once a new sensor reading is received, it is correlated to all the objects currently stored in the database. Previously detected objects are verified and the new objects are added into the database.

Feature recognition is easily grasped conceptually, but can be complicated to implement. In man-made areas, such as buildings, the range of shapes required to accurately depict the environment is limited, most areas can be approximated by simple shapes. In more natural settings, the variation in objects requires more complex shapes be defined. The larger dataset of shapes added to the algorithm and the increased complexity of the shapes requires large amount of computations. This in turn requires a very large amount of processing power.

With feature recognition, every shape must be classified, compared against the database of shapes, and stored for latter comparison. Cluttered environments will require a significant number of classifications and comparisons. As the vehicle finds more

objects, more processing power and time will be required. The variability in processing power requirements makes feature recognition difficult to implement on small scale UUVs.

Feature recognition also provides much finer detail than is required by the UUV for safe navigation. For safe navigation, the vehicle only needs to know that an object exists as a given position. It does not matter whether that object is cubic, cylindrical, or spherical. It only matters that the space is occupied. For that reason, occupancy grid mapping is a better choice.

B. OCCUPANCY GRIDS

Instead of recognizing and organizing objects by their shape and location, occupancy grids divide the area into a grid. The region to be mapped is defined and then tessellated. Each element, or cell, in the grid is assigned a value indicating whether the cell is occupied. This idea is easily extended from two-dimensional (checkerboard) to three-dimensions (stacks of cubes). Since the state of the cell could only take on one of two values. The cell is either empty or occupied. Since the cell can only be in one of two states, the probability of the cell being empty plus the probability of the cell being occupied must equal one (Equation 4-1).

$$P\left[s(C) = Empty\right] + P\left[s(C) = Occupied\right] = 1 \qquad 4\text{-}1$$

The information provided by the sensors does not directly map to the state of the cell. The state of the cell must be inferred from the sensor values. Because of this inference, the state of the cell is approximated. In an ideal world, the cell could only take on one of two values. The cell would either be empty or would be occupied. However, in practice, the sensors cannot provide the definitive state of the cell. Therefore, the probability of the state of each cell is recorded. The determination of the probability of the state of the cell given the current reading of the sensor is found using Bayes theorem Devore[14] defines Bayes Theorem in the following manner, 'Let A_1, A_2, …A_k be a

collection of k mutually exclusive and exhaustive events with prior probabilities $P(A_i)(i=1,\ldots,k)$. Then for any event B for which P(B)>0, the posterior probability of A_j given that B has occurred is

$$\mathrm{P}\left(A_j \mid B\right) = \frac{P\left(A_j \cap B\right)}{P\left(B\right)} = \frac{P\left(B \mid A_j\right)P\left(A_j\right)}{\sum_{i=1}^{k} P\left(B \mid A_i\right) \bullet P\left(A_j\right)} \qquad 4\text{-}2$$

Utilizing Bayes theorem will provide the probability of the state of the cell based on the current measurements only. What is needed is the state of the cell based on the current and all previous measurements. Elfes [16] provides an iterative solution to determining the probability that a cell is occupied given all previous measurements (Equation 4-3).

$$P\left[s(C) = Occ \mid \{r\}_{t+1}\right] = \frac{P\left[r_{t+1} \mid s(C) = Occ\right] * P\left[s(C) = Occ \mid \{r\}_t\right]}{\sum_{\forall s(C)} P\left[r_{t+1} \mid s(C)\right] * P\left[s(C) \mid \{r\}_t\right]} \qquad 4\text{-}3$$

r_{t+1}: The current measurement

$s(C)$: State of the cell

$\{r\}$: All measurement up to time t

Expanding the summation in the probability update equation yields Equation 4-4. From here, it is easy to see the four unknowns that must be determined for each update. The unknowns are listed in Table 7.

$$P\left[s(C) = Occ \mid \{r\}_{t+1}\right] = \frac{P\left[r_{t+1} \mid s(C) = Occ\right] * P\left[s(C) = Occ \mid \{r\}_t\right]}{P\left[r_{t+1} \mid s(C) = Empty\right] * P\left[s(C) = Empty \mid \{r\}_t\right] + P\left[r_{t+1} \mid s(C) = Occ\right] * P\left[s(C) = Occ \mid \{r\}_t\right]} \qquad 4\text{-}4$$

$P\left[s(C) = Occ \mid \{r\}_{t+1} \right]$	Probability that a cell is occupied given the current and all previous measurements
$P\left[s(C) = Occ \mid \{r\}_{t} \right]$	Probability that a cell is occupied given all the previous measurements
$P\left[s(C) = Empty \mid \{r\}_{t} \right]$	Probability that a cell is occupied given all the previous measurements
$P\left[r_{t+1} \mid s(C) = Occ \right]$	Probability of receiving the current measurement given that the cell is occupied
$P\left[r_{t+1} \mid s(C) = Empty \right]$	Probability of receiving the current measurement given that the cell is empty

Table 7. Definition of Terms

Since the cells can only have two states, empty or occupied, $P\left[s(C) = Empty \mid \{r\}_{t} \right]$ can be substituted with $1 - P\left[s(C) = Occ \mid \{r\}_{t} \right]$, reducing the number of values to be calculated for each cell. For example, assume a cell is an unknown state $P\left[s(C) = Occ \right] = 0.6$ and a new measurement is received. Based on the measurement and the corresponding probability functions the values of $P\left[r_{t+1} \mid s(C) = Occ \right]$ and $P\left[r_{t+1} \mid s(C) = Empty \right]$ are determined. Assume that $P\left[r_{t+1} \mid s(C) = Empty \right] = 0.2$ and $P\left[r_{t+1} \mid s(C) = Occ \right] = 0.4$. From these values the probability of the cell being occupied is updated (Equation 4-5).

$$P\left[s(C) = Occ \mid \{r\}_{t+1} \right] = \frac{0.4 * 0.6}{0.2 * (1 - 0.6) + 0.4 * 0.6} = 0.75 \qquad 4\text{-}5$$

The recursive nature of this problem requires a seed value for $P\left[s(C) = Occ \mid \{r\}_{t} \right]$. At time zero, or the time prior to any measurements, this value indicates all the information the vehicle has about its environment. When the vehicle has zero information about its environment the cells are initialized to 0.5. An initial value of 0.5 indicates that the cells are equally likely to be occupied or empty. As more information is attained this value is updated. This value will constantly be evolving based on the current input from the sensor.

The next values needed are the probability of receiving a particular measurement value given the state of the cell. These probability distributions are individual to the type of sensor employed. Since the probability distribution is dependent on the sensor type, the distributions should be defined prior to vehicle operation. The probability distributions associated with the sensor can be determined either theoretically or experimentally. Whether distributions are derived theoretically or through experimentation, their values can be stored and easily referenced at run time. The sensor probability distributions for the FLS were derived experimentally. The methods used to derive them will be covered in Chapter IV.

The typical occupancy grid update starts when a new measurement is received. The sensor location is correlated to an occupancy grid location. The current occupancy grid value is retrieved. Next, the measurement is converted into two separate probabilities, the probability of receiving the signal assuming that the location is empty and the probability of receiving the signal assuming the location is occupied. Utilizing Equation 4-4 the new value of the probability of the grid cell being occupied is calculated and stored. This process will repeat for every sensor measurement that is received.

C. COMBINING MULTIPLE SENSORS

The previous discussion only covered the case in which the grid was dependent on a single sensor. However, in most robotic applications, the vehicle will have multiple sensors. The vehicle may have many sensors of the same type or a multitude of different classes of sensors. It is through the combination of multiple sensors that a better understanding of the environment can be made. In order to take full advantage of all the sensors that the vehicle deploys, the occupancy grid map must account for all sensor values.

The most accurate method of combination would be to use a superbayesian approach. In this method, all sensors probabilities would need to include the dependency on the other sensors utilized. For example, when utilizing two different sensors, S1 and S2, the probabilistic sensor model for S1must also take into account any information provided by S2. The new probabilistic model for S1 would be the probability of

receiving the particular sensor value given that the cell is occupied and that S2 received a given sensor value. The same holds true for S2. This redefines the probabilistic model used for each sensor utilized.

There are two distinct disadvantages associated with the method. Many unmanned vehicles are capable of changing the sensors they employ. A vehicle that has five different sensors would require 120 (5!) sensor models to account for all of the possible sensor configurations. This alone makes this type of implementation impractical. Assuming that all 120 sensor models are created, if any sensor is added or upgraded all the previous sensor models are superseded. On today's vehicles that are capable of rapid changes in sensor configuration, this method is not practical.

An alternative method is to maintain separate occupancy grids for each sensor and then combine the grids utilizing an independent opinion pool. This method has proven successful by Elfes in [15]. In this scenario, each sensor will generate its own occupancy grid values. Assume the vehicle is utilizing sensors, S1 and S2, which in turn generate occupancy grids P1 and P2. The grid probabilities can be combined, as shown in Equation 4-6.

$$\text{Pooled Probability} = \frac{P_1 P_2}{P_1 P_2 + (1 - P_1)(1 - P_2)} \qquad 4\text{-}6$$

The individual occupancy grid values will be dependent only on the sensor input and its probabilistic model. Maintaining the independence between sensors allows for a singular sensor models to be created. For example, REMUS is also equipped with a side scan sonar. Defining independent models for side scan sonar and FLS allows for their inclusion or exclusion in the reconstruction process based on current task. For the previous vehicle with five sensor, this reduces the number of required sensor models from 120 to 5. With the fewer number of models required, it becomes practical to implement this algorithm on unmanned vehicles.

D. SUMMARY

Occupancy grids can organize multiple sensor inputs into a format that can be used for safe and accurate navigation. They do not require any prior knowledge of the

area and can easily be utilized in highly congested areas. Occupancy grids are also well suited to combine multiple sensor inputs. The ability to navigate without any prior knowledge and the ability to determine the environment from all sensor inputs is an ideal combination for UUVs.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS AND CONCLUSIONS

A. RESULTS

Before discussing the results, it is important to recall the objectives of the research. REMUS's ability to avoid obstacles is dependent upon its information about the environment. Currently REMUS is utilizing two-dimensional representation of the environment for path planning and obstacle avoidance. This limited REMUS's ability to navigate successfully in cluttered environments, like under a bridge.

In 2007, REMUS was deployed in the Charles River and sent on a path underneath the Massachusetts Avenue Memory Bridge. Figure 19 is a picture of the Massachusetts Avenue Bridge. Figure 20 shows the path of the vehicle. The path is shown in blue and the red circles indicate edges of the pylons supported the bridge. The GPS position of the pylons were acquired by placing a handheld GPS unit in close proximity to the part of the pylons that were above the surface. A limited number of satellites were acquired while receiving the position due to the bridge occluding GPS satellite reception. Due to the limited number of satellites received, there may be errors associated with the GPS positions of the columns.

Figure 19. Massachusetts Avenue Bridge, Boston, MA, from [18]

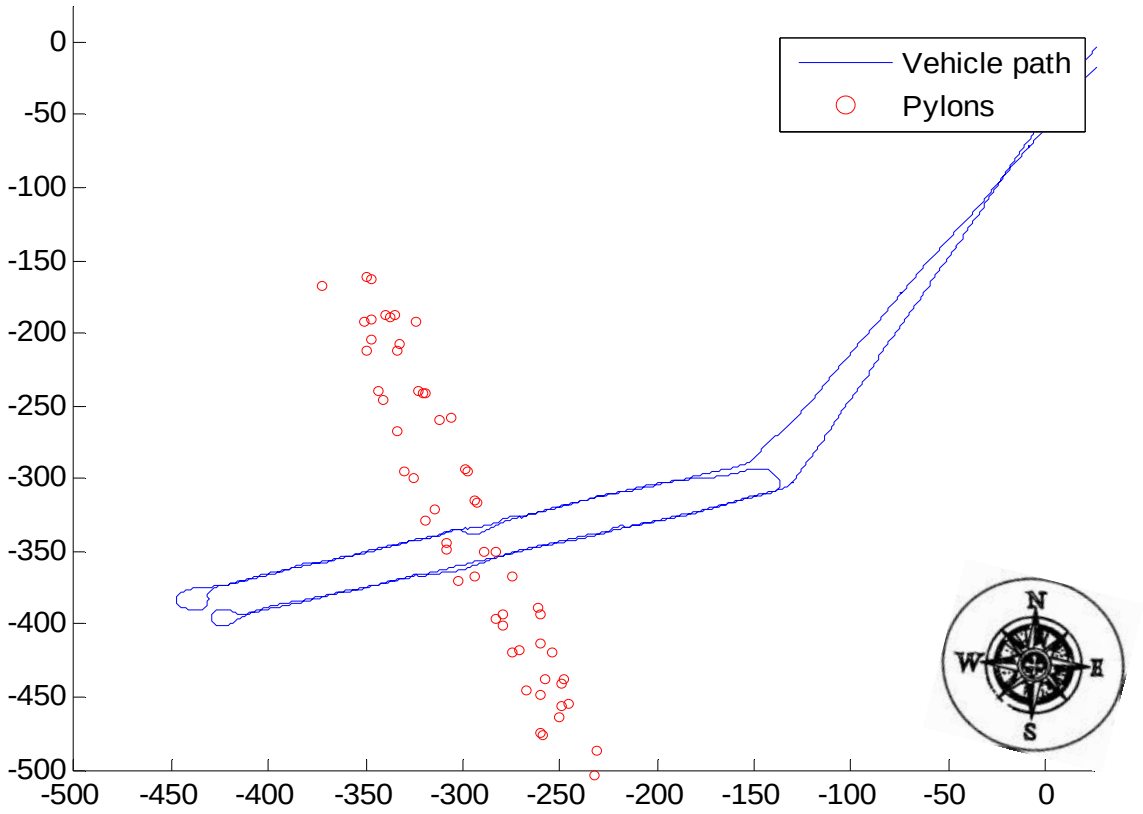

Figure 20. Vehicle Path in the Charles River and under the Massachusetts Avenue Bridge

As discussed in Chapter I, the vehicle would not be able to navigate successfully between the pylons and over the berm if it relied on obstacle avoidance and two-dimensional representations of the bridge. This can be seen by Figure 4. With the combination of horizontal and vertical FLS and occupancy grids, a path emerges. Figure 21 shows the occupancy grid created. The grid was formed using eleven images; 6 horizontal and 5 vertical. These were the total number of images available to the vehicle as it approached the bridge. Figure 21 is composed of small cubes. A solid black cube corresponds to a probability of occupancy of nine-tenths or greater. Cubes with a probability between five and nine tenths are increasing shades of red. Cubes with a probability of one half or less are not depicted. Cubes with values of one half or less indicate unknown areas or areas not likely to be occupied. These cubes are not displayed.

Comparing Figure 21 with Figure 4 the success of occupancy grids readily stands out. In the horizontal image of Figure 4, the four berms were apparent. The problem with the horizontal view was that the berm appeared as a solid wall. In Figure 21, the central section of the wall created by the bern has a reduced height. This section is the area ensonified by the vertical FLS. The vertical FLS has lowered the height of the berm to its correct value. The vehicle can now recognize the path through the berm. A similar effect happens with the space between the pylons. Due to the ambiguity of the vertical FLS, the columns formed a vertical wall that covered the entire field of view. However, once combined with the horizontal FLS information, the edges and dimensions of the columns become easily identifiable.

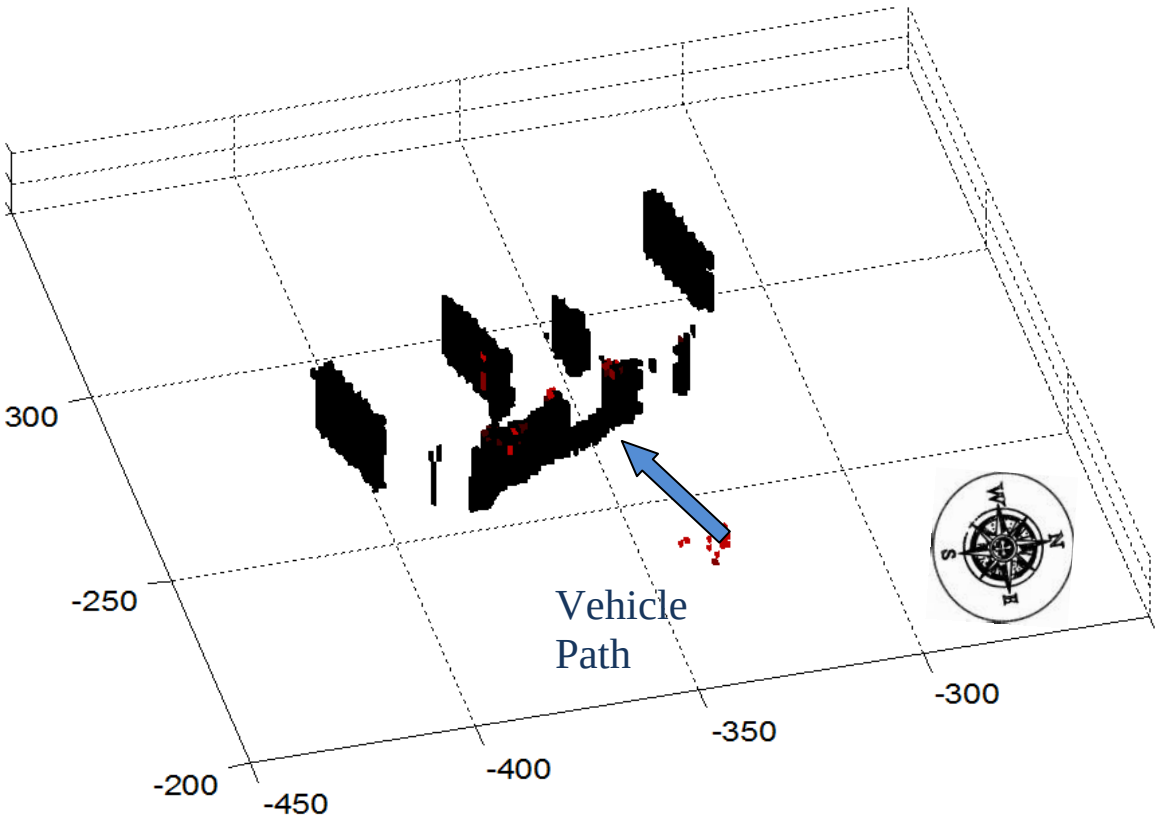

Figure 21. Occupancy Grid Developed with 11 consecutive images (6 Horizontal, 5 vertical) with the vehicle on a southwest approach

Successfully locating a path for the vehicle is just part of the problem; the vehicle must also be able to correctly place the path into the global space. To determine the

accuracy of the generated map, the positions of the generated pylons are compared to measured values. The GPS location of the pylons was determined with a hand held GPS unit. Figure 22 shows an aerial view of the occupancy grid with the GPS measurements overlaid in green circles. The distance between the GPS position and the closest gridded pylon location were calculated. The results are listed in Table 8.

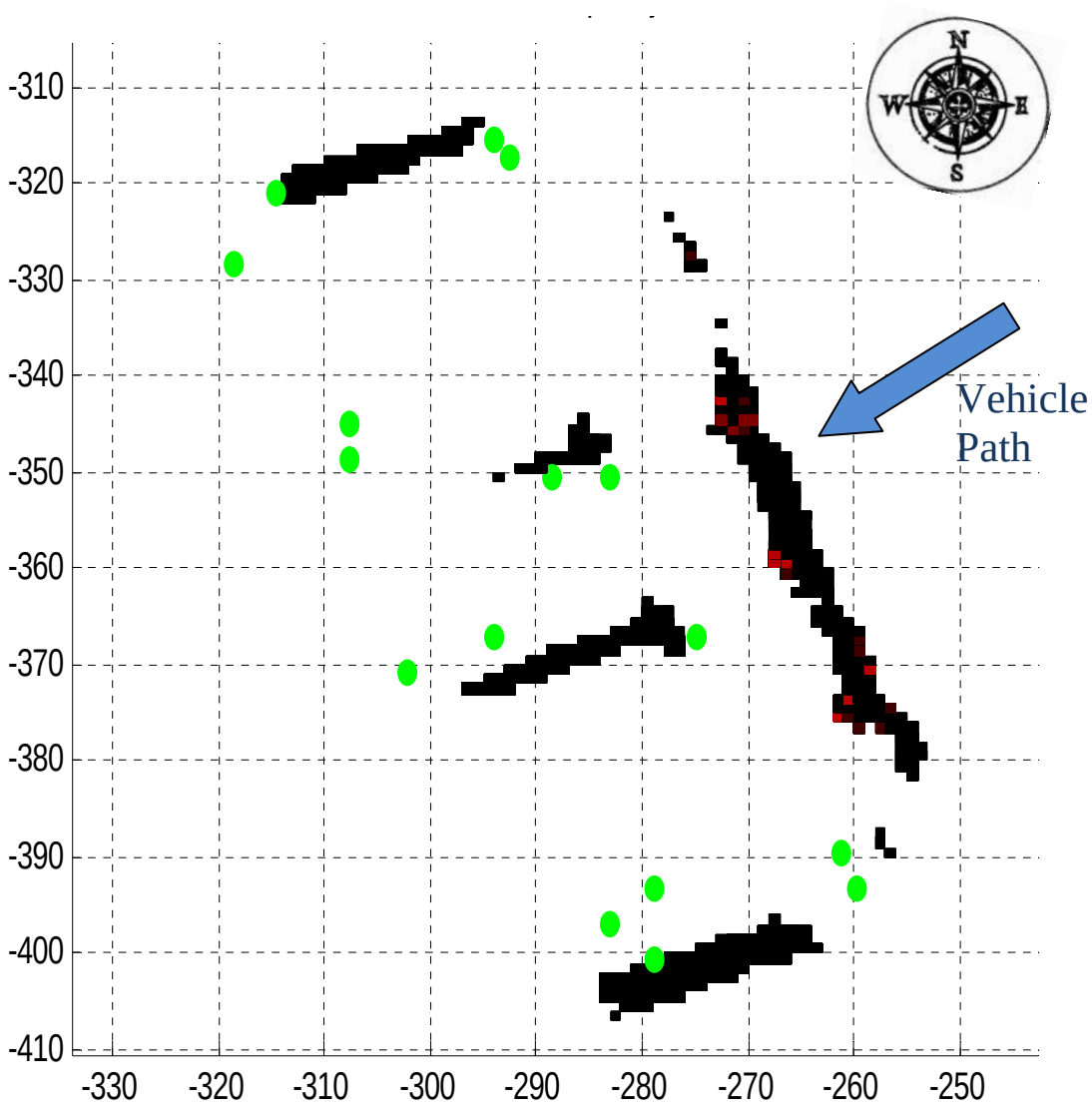

Figure 22. Occupancy Grid and Column Locations

Actual	Cell Location	Distance (m)
(-294.1, 315.2)	(-296, -315)	1.9
(-292.7, -317.1)	(-296, -316)	3.5
(314.6, -320)	(-314. -320)	0.6
(-288.6, -350.4)	(-289, -350)	0.6
(-283.1, -350.4)	(-284, -349)	1.7
(-274.9, -367.1)	(-276. -367)	1.1
(-294.1, -367.1)	(-291, -369)	3.6
(-261.2, -389.3)	(-264, -397)	8.2
(-259.8, -393)	(-267, -396)	7.8
(-279, -293)	(-273, -298)	7.8
(-283.1, 396.8)	(-281, -401)	4.7
(-279, -400.5)	(-279, -401)	0.5

Table 8. Actual Pylon Location versus Cell Location

When analyzing this data, a couple of observations become apparent. There is a significant offset between the location and gridded location of the lower pylon. The possible sources of error will be discussed in the following section. Excluding the lower most pylon, the rest of the data shows a mean distance error of 1.8 m. This error is much larger than expected with the high-resolution capabilities of the FLS.

B. SOURCES OF ERROR

Several factors play a role in the overall error distance. One of the main contributing factors is the grid size. The grid size (1m x 1m x 0.5m) was chosen to reduce the number of computations required. The grid size also provided enough

resolution for the vehicle to determine a path through under the bridge. However, this large grid size does not lend itself to a direct relationship with the sensor resolution.

Another contributing factor is the method used for pixel probability placement into the grid. A single grid cell will have multiple pixels plotted to it for a single update. In this experiment, the highest probability values were placed within the cell. This method disregards information provided by the range of pixel contained in the cell.

The final contributing factor to the error is the vehicle position and orientation at the time of the image. There are two separate contributors to this one source of error. First, the image data is not sampled at the same time or by the same computer as the vehicle state information. Because of this, the vehicle state information was interpolated to determine an approximate value at the time of the image. Any error in the state information is propagated through the algorithm and results in displaced detections. The second contributor to errors in the vehicle state is the approximations made by the internal navigation unit. The vehicle received a position fix at the beginning of the mission but was unable to receive any more fixes for the duration of the mission.

C. RECOMMENDATIONS

The research has demonstrated the proof of concept of utilizing dual FLS with occupancy grids to reconstruct a three-dimensional environment. However, progress needs to be made before the process is deployable on a vehicle in real time. Some of the areas that need further investigation are detailed below.

1. Probabilistic FLS Model

The validity of the model created for this experiment is limited to use in the Charles River. The model does not take into account the variability of many of the sources of noise. Many factors can affect the noise distributions. To be truly useful the noise distribution will need to take into account changes in the vehicles operational parameters. For example, how does the noise change as the speed of the vehicle changes? Does flow noise contribute significantly? What is the correlation in the noise model and the acoustic noise in the environment? Many parameters need to be analyzed

to develop a robust noise model. The detection distribution utilized for the FLS model will also need refinements. It is recommended that specific experiments be undertaken to provide a better classification of noise and detection distributions.

2. Algorithm Improvements

The algorithm in its current form is too slow to implement on an operational vehicle. An update based on a single image requires approximately 60 seconds to process. This is unacceptable for real time operation. It is recommended that the algorithm be transferred from a Matlab, a prototype environment, to a language designed for speed. One possible source of improvement is to down sample the FLS images. The current images provide a bearing resolution of 0.044 degrees. The actual minimum resolution of the FLS is specified as one degree. If the images are re-sampled in a manner to minimize the data lost, the number of computations can be drastically reduced. One possible method that can be used to resample the images is to use a Markov Random Field to determine the relationship among the pixels. Once a viable method for resampling has been found, the corresponding distribution functions will need to be created.

3. Combine the Occupancy Grid Route Planning Algorithm

The final recommendation is to couple this algorithm with route planning. The main goal of this thesis is to provide the vehicle a method to determine safe routes for navigation in a cluttered environment. While this thesis has shown a feasible method, until it is coupled with a route planning algorithm, its full utility cannot be determined.

D. CONCLUSIONS

There are issues to be addressed and further research needs to be conducted, but this research can stand as a proof of concept. It is possible to provide UUVs with a reconstructed three-dimensional representation of the underwater environment. This representation can then be used for route planning and obstacle avoidance. Using a

three-dimensional reconstruction provides UUVs with the capability of successfully navigating in cluttered environments. With this capability, it becomes possible to safely deploy UUVs into more complex environments.

APPENDIX: MATLAB CODE

A. MERGING VEHICLE AND SONAR IMAGE DATASETS

```
dataset = zeros(8100,15);
% %
% dataset = [elapsed time, lat, lon, x, y, depth, altitude, heading, pitch, ...
%                 roll, velocity, headingRate, Horizontal Image #, Vertical
Image #]
% Original Data located in VehicleData, VertData, HorData

%Vehicle Elapsed time AtBeginning of first turn:    736 sec
%Sonar Elapsed time at Beginning of first turn:     685 sec

offset = 736-685
j = 1;
hd1 = zeros(1,15);
hd2 = hd1;
vd1 = zeros(1,15);
vd2 = vd1;
for i = 35:3000
    %Retrieve time of Horizontal FLS
    Htime = HorizData(i,1);
    %Retrieve time of Vertical FLS
    Vtime = VertData(i,1);

%Get Vehicle State Associated with Horizontal FLS and Interpolate
    Hn = find(VehicleData(:,9)<(Htime+offset), 1, 'last')-1;
    hd1 = [VehicleData(Hn,9), VehicleData(Hn, 10:11), 0,
0,VehicleData(Hn,12:18)];
    Hn = Hn+1;
    hd2 = [VehicleData(Hn,9), VehicleData(Hn, 10:11), 0,
0,VehicleData(Hn,12:18)];
    intp = (hd2(1)-(Htime+offset))/(hd2(1)-hd1(1));
     [Hn, VehicleData(Hn-1, 9), Htime+offset, VehicleData(Hn,9), intp]
    hd = hd1 + intp*(hd2-hd1);
%Get Vehicle State Associated with Vertical FLS and Interpolate
    Vn = find(VehicleData(:,9)<(Vtime+offset), 1, 'last');
    vd1 = [VehicleData(Vn,9), VehicleData(Vn, 10:11), 0,
0,VehicleData(Vn,12:18)];
    Vn = Vn+1;
    vd2 = [VehicleData(Vn,9), VehicleData(Vn, 10:11), 0,
0,VehicleData(Vn,12:18)];
    intp = (vd2(1)-(Htime+offset))/(vd2(1)-vd1(1))
    vd2 = vd1+intp*(vd2-vd1);
 %Store combined data to a new dataset ordered by offset time
    switch logical(true)
        case(Htime<=Vtime)
            dataset(j,:) = [hd2, i, i-1, 0];
            dataset(j+1, :) = [vd2, i, i, 0];
            j  = j+2;
        case(Htime>Vtime)
            dataset(j,:) = [vd2, i-1, i, 0] ;
            dataset(j+1, :) = [hd2, i , i, 0];
            j = j+2;
    end
end

%Convert GPS to meters in a local grid
```

```
% Reference point (42.358301, -71.088055)
for i = 1:size(dataset,1)
    lat = dataset(i, 2);
    lon = dataset(i,3);
    dataset(i,4) = (lat-42.358301)*60*1853.47;
    dataset(i,5) = cos(2*pi*lat/360)*60*1853.47*(lon--71.088055);
end

clear hd1 hd2 vd1 vd2 intp i Vn Hn Htime Vtime j lat lon
% clear VehicleData VertData HorizData
```

B. GENERATING FLS PROBABILITY MODEL

```
bins = linspace(0,4000,25);
binstorage = zeros(size(bins));

%Generate Noise Distributions
binstorage = zeros(461, 2048, numel(bins));
for i = 300:20:600  %Range of Images involved in Noise Distribution
    %Combine 20 Images
    for j = 1:20
        imageData(:,:, j) = OpenHorSonarImage('..\SonarImages\8_bit',
'fls_horz_',  4, '.png', i+j-1);
    end
    m = zeros(size(binstorage));
    %Create Histograms
    for k = 1:size(imageData,1)
        for p = 1:size(imageData,2)
            m(k,p,:) = hist(imageData(k,p,:),bins);
        end

    end
    binstorage = binstorage+m;
    i = i
    clear imageData

end

clear i j k p m b1
%Save Histogram as Noise Histogram
Vnoise = binstorage;

%Generete Sensor Histograms
imageData = zeros(461, 2048, 20);
binstorage = zeros(461, 2048, numel(bins));
%Image Ranges used in Dataset
I_range = [820, 960; 1125, 1265; 1505, 1617; 1840, 1910];
for l = 1:4
    start = I_range(l,1);
    stop = I_range(l,2);
    for i = start:20:stop
        for j = 1:20
         imageData(:,:, j) = OpenHorSonarImage('..\SonarImages\8_bit',
'fls_horz_',  4, '.png', i+j);
        end

        m = zeros(461, 2048, numel(bins));
```

```
        for k = 1:size(imageData,1)
            for p = 1:size(imageData,2)
                m(k,p,:) = hist(imageData(k,p,:),bins);
            end
        end
        binstorage = binstorage+m;
        i = i
    end
end
 %Save Sensor Distribution
Vsensor = binstorage;
Vbins = bins;
clear imageData binstorage I_range start stop
clear i j m k p
%

%Remove Noise From Sensor Distribution
n = zeros(size(Vbins));
s = n;
t = n;
i = 200; j = 880;
%Update the Distributions for Each Pixel
% for i = 1:461
%     for j = 1:2048
        %Retrieve Noise Histogram
        n(:) = Hnoise(i,j,:);
      figure(4); subplot(4,2,1); bar(Hbins, n); title('Noise
Distribution')
        %Normalize Noise
        n = n/sum(n);
        [mn, mi] = max(n);
      figure(4); subplot(4,2,2); bar(Hbins, n);title('Normalized Noise
Distribution, n')
        % Retieve Sensor Histogram
        s(:) = Hsensor(i,j,:);
      figure(4); subplot(4,2,3); bar(Hbins, s); title('Detection
Distribution')
        % Normalize sensor over range of Noise Distribution
        s = s/sum(s.*(n>0));
      figure(4); subplot(4,2,4); bar(Hbins, s); title('Normalized
Detection Distribution, s')
        %Remove Noise and Ensure all values are positive, then Normalize
        s = s-n;
%                figure(4); subplot(4,2,5); bar(Hbins, s); title('s-n')
        s= s + abs(min(s.*(s<0)));
        s = s/sum(s);
                figure(4); subplot(4,2,6); bar(Hbins, s); title('(s-n)
raised to a min of zero')
%         s = s+0.001*abs(min(s+10.*(s==0)));
        figure(4); subplot(4,2,7); bar(Hbins, (s+n)/sum(s+n));
title('Adjusted Detection Distrobution')
%         s = (1-s).*(s~=0);
        s = s/sum(s);
                figure(4); subplot(4,2,8); bar(Hbins, (s+n)/sum(s+n));
title('Adjusted Detection + Noise')
        n = n+max(min(n+10.*(n==0)),0.01);

    Save Adjusted Distribution to Sensor and Noise Models
        Hnoise(i,j,:) = n/sum(n);
        Hsensor(i,j,:) =s;
    end
end
```

C. GENERATING THE MAP

1. Main Function

```
%ESTABLISH THE BOUNDARIES OF THE GRID
b = [0 0.25 50; 0 .25 50; 0 0.25 10];
b(1,1:3) = b(1,1:3) + -300;
b(2,1:3) = b(2,1:3) + -385;
b(1:2, 2) = 0.25;
for i = 1:3
    b(i,4) = numel(b(i,1):b(i,2):b(i,3));
end
b = b

%Initialize all variables
%Image Containers
imgH = zeros(461, 2048);
imgS = zeros(size(imgH));
imgV = zeros(461, 1024);
imgSv = zeros(size(imgV));

%Grid Containters
gH = 0.5*ones(b(1,4), b(2,4), b(3,4));
gV = 0.5*ones(size(gH));
gN = zeros(size(gH));
gS = zeros(size(gH));
gM = 0.5*ones(size(gH));

%Vehicle State
vd = zeros(1,6);

HIN = 0;
VIN = 0;

dr = [1662, 1672 ; 2280, 2290 ]  %Establish images be considered
for l =1
for vdn = dr(l,1):dr(l,2)
    tic
    %Extract Vehicle state and Images Numbers
    vd = [dataset(vdn, 5), dataset(vdn, 4), dataset(vdn, 7), 90-
dataset(vdn, 8), dataset(vdn, 9), dataset(vdn, 10)];
%   Generate Horizontal Map
    if HIN < dataset(vdn, 13)
        HIN = dataset(vdn, 13);
        imgH = OpenHorSonarImage('../SonarImages/RTheta', 'fls_horiz_',
4, '.pgm', HIN);  %Get Images
        imgH(find(isnan(imgH)))=2;
        imgS = ProbImages(imgH, Hsensor, Hbins);        %Detection Prob
Image
        img = ProbImages(imgH, Hnoise, Hbins);          %Noise Prob Image
        gS = HorImage2Map(imgS, vd, b);          %Detection Prob Grid
        gMask = gS>=0;                                  %Mask of updated
Values
        gN = HorImage2Map(img, vd, b);           %Noise Prob Grid
        %Horizontal Grid Update
        gS = (gS.*gH)./(gN.*(1-gH)+gH.*gS);
        gH = gS.*gMask+gH.*(~gMask);
    end

%   Generate Vertical Map
    if VIN < dataset(vdn, 14)
```

```
        VIN = dataset(vdn, 14);
        imgV = OpenVertSonarImage('..\SonarImages\RTHeta', 'fls_vert_',
4, '.pgm', VIN);  %Get Image
        imgSv = ProbImages(imgV, Vsensor, Vbins);   %Detection Prob Image
        imgV = ProbImages(imgV, Vnoise, Vbins);     %Noise Prob Image
        gS = VertImage2Map(imgSv, vd, b);           %Detection Prob Grid
        gMask = gS>=0;                              %Grid Mask
        gN = VertImage2Map(imgV, vd, b);            %Noise Prob Grid
        % Vertical Grid Update
        gS = (gS.*gV)./(gN.*(1-gV)+gS.*gV);
        gV = gS.*gMask+gV.*(~gMask);
    end


%   Pooled UPdate
    gM = (gV.*gH)./(gH.*gV+(1-gV).*(1-gH));

    [HIN, VIN, vdn]  %Show Loop Progress
end
end
```

2. Converting FLS Images to Probabilities

```
function Prob = ProbImages(img,  SensorDist, binSpacing)
%This function creates to probability images based upon the
provided image
%and the Sensor Probability Distribution (modeled as a histogram).
Each
%bin contains the probability of recieving a sensor value within
the edges
%of the bin.
%
% %Inputs:
%     img:  Sensor Image    M x N
%     binSpacing: Vector indicating the edges of the probability
bins  (length P)
%     SensorDist:  Probability Distribution of the sensor.
%            M x N x P
%   Outputs:
%     Prob:  An M x N matrix where each element represents the
       probability of receiving each pixel

%
Prob = 0.5*ones(size(img));

%Traverse the entire image
for j = 1:size(img,2)
    for i = 1:size(img,1)
       %Find out which Bin number the intensity falls into
      binNum = find(img(i,j)<binSpacing, 1, 'first');
      if binNum >0
       %Retrieve probability of receiving that intensity value

         Prob(i,j) = squeeze(SensorDist(i,j,binNum));
       else
         Prob(i,j) = 0.0001;  %Assign a minimum Probability
      end
    end
end
```

3. Mapping Image Data to Grid Locations

a. Horizontal FLS

```
function imageMap = HorImage2Map(img,VehicleData, bounds)

% %bounds
%   [ xmin, xres, xmax, # of cells
%     ymin, yres, ymax, # of cells
%     zmin, zres, zmax, # of cells  ]
temp = [];
cnt = 1;

ambspacing = 0.5*pi/180;
ambend = 7*pi/180;

%Set Vehicle position and orientation parameters
x0 = VehicleData(1);  %LocalX
y0 = VehicleData(2);  %LocalY
z0 = VehicleData(3);  %Depth
heading = VehicleData(4)*pi/180;
pitch = VehicleData(5)*pi/180;
roll = VehicleData(6)*pi/180;


%Gen imageMap
xr = bounds(1,1):bounds(1,2):bounds(1,3);
yr = bounds(2,1):bounds(2,2):bounds(2,3);
zr = bounds(3,1):bounds(3,2):bounds(3,3);
imageMap = -1*ones(bounds(1,4), bounds(2,4), bounds(3,4));

%Establish Image Parameters
imgH = size(img,1);
imgW = size(img,2);
%Spherical Coordinates
theta = linspace(pi/4, -pi/4, imgW);
r = linspace(90,0, imgH);
phi = -ambend:ambspacing:ambend;

for phiI = 1:size(phi,2)

    [gr, gt, gphi ] = ndgrid(r, theta, phi(phiI));
    %Convert data from Local Spherical to Global Cartesian
    [x, y, z] = sph2cart(gt+heading, gphi+pitch, gr);
    %Move Image Data to Vehicle Location
    x = x+x0;
    y = y+y0;
    z = z+z0;
    %Covert Image Position to Grid Values
    xi = ceil( (x-bounds(1,1))/bounds(1,2));
    xi = ((xi>0)&(xi<size(xr,2))).*xi;
    yi = ceil( (y-bounds(2,1))/bounds(2,2));
    yi = ((yi>0)&(yi<size(yr,2)-1)).*yi;
    zi = ceil( (z-bounds(3,1))/bounds(3,2));
    zi = ((zi>0)&(zi<size(zr,2)-1)).*zi;

    %Store Image Values into Grid Position
    for rI = 1:imgH
      for thetaI = 1:imgW
```

```
          %Determine if image value is contained within map
          valid = (xi( rI, thetaI, 1)>0 & yi( rI, thetaI, 1)>0 &
zi( rI, thetaI, 1)>0);
          %If Valid Store Data contain in image pixel to
corresponding Grid
          %Locations
          if valid
                val = imageMap(xi( rI, thetaI, 1), yi( rI, thetaI,
1), zi( rI, thetaI, 1));
                imageMap(xi( rI, thetaI, 1), yi( rI, thetaI, 1),
zi( rI, thetaI, 1)) = max(img(rI, thetaI), val);
            end

        end
    end
end
```

b. Vertical FLS

```
function imageMap = VertImage2Map(img,VehicleData, bounds)
% %bounds
%   [ xmin, xres, xmax
%     ymin, yres, ymax
%     zmin, zres, zmax  ]
temp = [];
cnt = 1;

ambspacing = 0.5*pi/180;
ambend = 7*pi/180;

%Set Vehicle position and orientation parameters
x0 = VehicleData(1);  %LocalX
y0 = VehicleData(2);  %LocalY
z0 = VehicleData(3);  %Depth
heading = VehicleData(4)*pi/180;
pitch = VehicleData(5)*pi/180;
roll = VehicleData(6)*pi/180;


%Gen imageMap
xr = bounds(1,1):bounds(1,2):bounds(1,3);
yr = bounds(2,1):bounds(2,2):bounds(2,3);
zr = bounds(3,1):bounds(3,2):bounds(3,3);
imageMap = -1*ones(size(xr,2), size(yr,2), size(zr,2));

imgH = size(img,1);
imgW = size(img,2);
phi = linspace(-pi/16, pi/16, imgW);
r = linspace(90,0, imgH);
theta = -ambend:ambspacing:ambend;

for thetaI = 1:numel(theta)
    [gr, gt, gphi ] = ndgrid(r, theta(thetaI), phi);
    [x,y,z] = sph2cart(gt+heading, gphi+pitch, gr);
    x = x+x0;
    y = y+y0;
    z = z+z0;

    xi = ceil( (x-bounds(1,1))/bounds(1,2));
    xi = ((xi>0)&(xi<size(xr,2))).*xi;
    yi = ceil( (y-bounds(2,1))/bounds(2,2));
```

```
    yi = ((yi>0)&(yi<size(yr,2)-1)).*yi;
    zi = ceil( (z-bounds(3,1))/bounds(3,2));
    zi = ((zi>0)&(zi<size(zr,2)-1)).*zi;
%     size(imageMap)
%     [size(xr, 2), size(yr,2);max(max(max(xi))),
min(min(min(xi))); max(max(max(yi))),
min(min(min(yi)));max(max(max(zi))), min(min(min(zi)))]
    for rI = 1:imgH
      for angle = 1:imgW
%           temp = [xi(rI, thetaI, 1), yi(rI, thetaI, 1),zi(rI,
thetaI, 1)]
          valid = (xi( rI, 1, angle)>0 & yi( rI, 1, angle)>0 & zi(
rI, 1, angle)>0);
          if valid
                val = imageMap(xi( rI, 1, angle), yi( rI, 1,
angle), zi( rI, 1, angle));
                imageMap(xi( rI, 1, angle), yi( rI, 1, angle), zi(
rI, 1, angle)) = max(img(rI, angle), val);
            end

        end
    end
end
%Set portions of the map without data to a value of .5 (unknown
state).
% imageMap = (imageMap <0).*.5 + imageMap.*(imageMap>=0);
% imageMap = reshape(imageMap, bounds(4,1)*bounds(4,3),
bounds(4,2));
```

4. Display the Grid

```
function drawGrid(gridMap, b, axis_handle)
%Generic Cube of Cell Dimensions
xcube = b(1,2)*  [ 0 0 1 1 0 NaN 0 1 NaN 1 0;...
            0 0 1 1 0 NaN 0 1 NaN 1 0];
xcube = xcube+b(1,1);
ycube = b(2,2)*  [ 0 1 1 0 0 NaN 1 1 NaN 1 1;...
            0 1 1 0 0 NaN 0 0 NaN 0 0];
ycube = ycube+b(2,1);
zcube = b(3,2)*  [ 1 1 1 1 1 NaN 1 1 NaN 0 0;...
            0 0 0 0 0 NaN 1 1 NaN 0 0];
zcube = zcube +b(3,1);
%Traverse entire Grid,  Draw cubes based on the Probability contained in
%the cell
for i = 1:size(gridMap, 1);
    for j = 1:size(gridMap, 2);
        for k = 1:size(gridMap, 3);

            switch logical(true)
                case gridMap(i,j,k) >0.95
                    surf(xcube+i*b(1,2), ycube+j*b(2,2), zcube+k*b(3,2),
'facecolor', [0 0 0], 'edgecolor', 'none');
                case gridMap(i,j,k) >= 0.8
                    surf(xcube+i*b(1,2), ycube+j*b(2,2), zcube+k*b(3,2),
'facecolor', [0.25 0 0], 'edgecolor', 'none' );
                case gridMap(i,j,k) > 0.7
                    surf(xcube+i*b(1,2), ycube+j*b(2,2), zcube+k*b(3,2),
'facecolor', [0.5 0 0], 'edgecolor', 'none');
                case gridMap(i,j,k) > 0.5
                    surf( xcube+i*b(1,2), ycube+j*b(2,2), zcube+k*b(3,2),
'facecolor', [0.75 0 0], 'edgecolor', 'none');
            end
            hold on
        end
    end
    if mod(i,10) == 0
        i = i
    end
end
axis([ b(1,1) b(1,3) b(2,1) b(2,3) b(3,1) b(3,3)]);
```

THIS PAGE INTENTIONALLY LEFT BLANK

WORKS CITED

[1] D. P. Horner and A.J. Healey, S. Kragelund, AUV Experiments in Obstacle Avoidance. Proceedings of the IEEE Oceans 2005 Conference, September 2005.

[2] T. Furukawa, Reactive Obstacle Avoidance for the REMUS Autonomous Underwater Vehicle Utilizing a Forward Looking Sonar. *M.S. Thesis Naval Postgraduate School*, 2006.

[3] K. Nagatani, H. Ishida, S. Yamanaka and Y. Tanaka, Three Dimensional Localization and Mapping for Mobile Robot in Disaster Environments. *Proceedings of the 2003 IEEE/RSJ International Conference*, pp 3112-3117. 2003.

[4] D. Langar and M. Herbert, Building Qualitative Elevations Maps from Side Scan Sonar Data for Autonomous Underwater Navigation, *Proceedings from 1991 Robotics and Automation International Conference*, pp 2478-2483. 1991.

[5] U.S. Navy, UUV Master Plan, November 2009.

[6] Hydroid Inc., Homepage [www.hydroidinc.com], February 2009.

[7] A. Healey and D. Marco, Command, Control and Navigation Experimental Results with the NPS ARIES AUV. *IEEE Journal of Oceanic Engineering – Special Edition. 2001.*

[8] M. Dolbec, Velocity Estimation Using Forward Looking Sonar. *MS Thesis Naval Postgraduate School*, 2007.

[9] T. Prestero, Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle. *M.S. Thesis Massachusetts Institute of Technology*, 2001.

[10] A. Washburn, A Short Introduction to Kalman Filters. Naval Postgraduate School 2007.

[11] W. Alameda, SEADeVIL A Totally Integrated Inertial Navigation System (INS) Solution. *Underwater Intervention Symposium*, 2002.

[12] Kearfott Corporation, Homepage [www.kearfott.com]. March 2009

[13] Introduction to Optics, Frank L. Pedrotti & Leno S. Pedrotti, Prentice Hall, (Frank L. Pedrotti, 1992).

[14] J. Devore, Probability and Statistics For Engineering and The Sciences, Thomson Brooks Cole, 2004.

[15] R. Lee Thompson, J. Seawall and T. Josserand, Two Dimensional and Three Dimensional Imaging Results Using Blazed Arrays, Applied Research Laboratories. University of Texas.

[16] A. Elfes, A Tesselated Probabilistic Representation for Spatial Robot Perception and Navigation. Carnegie-Mellon University. January 1989 .

[17] A. Elfes and L. Matthies, Sensor Integration for Robot Navigation: Combining Sonar and Stereo Range Data in a Grid-Based Representation. *Proceedings 26th IEEE Conference on Decision and Control.* 1987.

[18] Wikipedia Commons, Homepage [www.wikipedia.com], March 2009.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Doug Horner
   Naval Postgraduate School
   Monterey, California

4. Roberto Cristi
   Naval Postgraduate School
   Monterey, California

5. Don Brutzman
   Naval Postgraduate School
   Monterey, California